

UNIVERSIDAD CARLOS III DE MADRID

Escuela Politécnica Superior - Leganés

DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA



PROYECTO FIN DE CARRERA

**Módulo de Identificación Biométrica Mediante Huellas
Dactilares para Sistemas Empotrados**

AUTOR: RUBÉN JUSTE MECO

TUTOR: DR. RAÚL SÁNCHEZ REÍLLO

Leganés, 2010

TITULO: Módulo de Identificación Biométrica Mediante Huellas
Dactilares para Sistemas Empotrados

AUTOR: RUBÉN JUSTE MECO

TUTOR: DR. RAÚL SÁNCHEZ REÍLLO

La defensa del presente Proyecto Fin de Carrera se realizó el día 11 de Junio de 2010;
siendo calificada por el siguiente tribunal:

PRESIDENTE: Luis Mengibar Pozo

SECRETARIO: Michael García Lorenz

VOCAL: Fernando García Fernández

Habiendo obtenido la siguiente calificación:

CALIFICACIÓN:

Presidente

Secretario

Vocal

A mis Padres

A mi Mujer

Agradecimientos

En este documento que pone fin a esta etapa universitaria, quiero reflejar mi agradecimiento a:

Raúl Sánchez Reíllo, por confiar en mí, permitiéndome una vez mas trabajar con él.

A todos mis compañeros con los que he compartido horas de clase, de laboratorio, de biblioteca y por supuesto de cafetería, pues todos me han ayudado a conseguir nuestro objetivo.

A todos mis amigos, que por fin podrán dejar de preguntarme por mi Proyecto Fin de Carrera. Especialmente a Luis Alberto y Pablo, que me han enseñado mucho y sin ellos, difícilmente hubiera visto la “huella”.

A mi familia en general, especialmente a mi abuelo Faustino que espera este día tanto como yo, y a mis sobrinos Lucia, Christian y Manolo.

A mi Mujer Eva, que desde hace muchos años, cuando este objetivo parecía imposible siempre tuvo fe, porque por muy difícil que pueda parecer, con paciencia y esfuerzo los anhelos se consiguen. A nosotros ya nos quedan menos.

A mis padres, Juan Luis y Paquita, porque han sabido educarme, guiarme y aconsejarme, por los esfuerzos que han realizado por mi y por mi hermana, pues lo han hecho muy bien; especialmente a mi padre pues en el momento adecuado supo cimentar sólidamente mi conocimiento, sin lo que jamás hubiera ni siquiera haber iniciado esta aventura, y aun hoy continua enriqueciéndolo. Espero que estéis orgullosos.

Resumen

La seguridad es uno de los temas que más preocupan en la actualidad, pero no solo en lo referente a la integridad física del individuo, sino también en lo concerniente a la protección de sus bienes o datos.

En la sociedad actual se hace uso de multitud de servicios electrónicos para disfrutar de los bienes, e-banking, compras mediante el uso de tarjetas de crédito, e-mail, e-shopping; y dispositivos electrónicos para acceder a los datos, PCs, PDAs, Teléfonos Móviles y demás. Sin una seguridad eficiente estos servicios se ven comprometidos en su mayoría por sus propias vulnerabilidades o ataques de hackers, para evitarlos, la seguridad se debe entender como la garantía de que el individuo que hace uso de estos servicios o dispositivos es quien dice ser y pueda ser identificado o autorizado de forma única, esto debe hacerse de forma fiable, amigable, rápida y a un coste razonable.

Tradicionalmente, las personas se han identificado ante personal de control de acceso mediante pasaportes o carnets, que disponen de foto y firma, o han accedido a sus datos a través de passwords o claves de identificación personal, PIN (Personal Identification Number), siendo ambos métodos susceptibles de ser robados, suplantados y olvidados. Todos los esfuerzos destinados a proporcionar una seguridad eficiente se centran en la Biometría. Biometría es el procedimiento de reconocimiento humano basado en características únicas e inalterables del individuo, es decir, la biometría se fija en “Quien” es el individuo, no en lo que posee o conoce.

Los sistemas biométricos ofrecen métodos de identificación o autenticación de individuos, de forma económica, fiable y automática. Estos sistemas tratan de identificar al individuo a través de medidas de sus características físicas o conductuales, se trata básicamente de un sistema de reconocimiento de patrones, en general se extrae una muestra de estas medidas en un formato estandarizado, que se compara con una plantilla almacenada en un sistema de seguridad que ha sido obtenida previamente utilizando los mismos estándares y se ha establecido como la única identidad del individuo. Una estrecha relación entre la muestra y la plantilla confirma la identidad de la persona.

Las principales modalidades biométricas capaces de identificar a un individuo de forma precisa son, la voz, la retina, el iris, geometría de la mano, la cara y las huellas dactilares. Aunque las líneas de trabajo pretenden desarrollar sistemas de reconocimiento electrónico basados en todos o en un conjunto de estos biométricos, este texto se centra en las huellas dactilares, pues cada una de ellas es distinta, permanente y uno de los identificadores biométricos mas aceptados.

Los dispositivos capaces de capturar los datos biométricos, se conocen como sensores biométricos. Un sensor de huella dactilar lee la superficie del dedo, distinguiendo entre crestas y valles, y convierte la lectura analógica en digital utilizando un conversor analógico-digital (ADC).

El objetivo de este proyecto es obtener una imagen de huella dactilar de calidad utilizando un sensor de huella dactilar de desplazamiento, concretamente el AT77C104B fabricado por Atmel bajo tecnología de dispositivos de estado solido en su variante térmica, pues su reducido tamaño permite su integración en sistemas empotrados basados en microcontrolador como PDAs, Notebooks, Netbooks o Teléfonos Móviles, siendo el empleado en este estudio un ARM7TDMI-S de 32 bits, el LPC2292 de NXP.

Glosario

A

AFIS: Sistema biométrico automatizado para identificación de huella dactilar que compara un registro de huellas dactilares con los registros de una base de datos para determinar la identidad de un individuo.

AFAS: Sistema biométrico automatizado para la verificación de una huella dactilar que decide si la huella dactilar pertenece al individuo especificado.

Algoritmo: Conjunto ordenado y finito de instrucciones que permite a un sistema informático hallar la solución de un problema.

Arco: Patrón de huellas dactilares en el cual las crestas entran por un lado, se elevan en el centro y salen por el lado opuesto. Este patrón no presenta deltas o lazos.

B

Base de Datos: Conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

Bifurcación: Punto en una huella dactilar donde una cresta se divide formando dos crestas.

Biometría: Consiste en la identificación de un individuo a través del análisis de sus características físicas o rasgos conductuales.

Biométrico: Alguna de las características morfológicas o fisiológicas de un individuo.

C

Captura: Proceso de recopilación de una muestra biométrica de un individuo por medio de un sensor.

Captura en Vivo: Por lo general, se refiere a un dispositivo de captura de huellas dactilares que captura las imágenes electrónicamente por medio de un sensor.

Característica Conductual: Característica biométrica aprendida y adquirida con el tiempo, y no basada en la biología.

Características Físicas: Característica biométrica innata e inalterable.

Clave de Identificación Personal: Método de seguridad basado en el conocimiento.

Codificación: Es el proceso por el cual la información de una fuente es convertida en símbolos siguiendo las reglas de un código.

Comparación: Proceso de comparación de una referencia biométrica con una referencia o referencias almacenadas con anterioridad, para tomar una decisión sobre identificación o verificación.

Conveniencia: Parámetro para determinar lo conveniente que resulta un sistema biométrico.

Cresta: Líneas de relieve de la huella dactilar.

Cresta Independiente: Cresta de poca longitud mayor que una isla.

Cruce: Cresta que une dos crestas paralelas.

D

Dieléctrico: Materiales que no conducen la electricidad, por lo que se pueden utilizar como aislantes eléctricos.

E

e-banking: Banca electrónica, es la banca a la que se puede acceder mediante Internet.

e-mail: Servicio de red que permite a los usuarios enviar y recibir mensajes rápidamente mediante sistemas de comunicación electrónicos

e-shopping: Compras por Internet.

Equilibrio Térmico: Si entre dos cuerpos en contacto térmico no existe flujo de calor.

Espolón: Cresta que se bifurca y una de las ramas tiene una longitud muy pequeña.

Extracción: Proceso de conversión de una muestra biométrica capturada en datos biométricos para que puedan ser comparados con una referencia.

F

Fotodiodo: Es un semiconductor construido con una unión PN, sensible a la incidencia de la luz visible o infrarroja.

H

Huella Dactilar: Es la impresión visible o moldeada que produce el contacto de las crestas papilares

Huella Dactilar Latente: Imagen de una huella dactilar que queda en una superficie tocada por un individuo.

Hacker: Criminal informático.

I

Identificación: Tarea en la cual el sistema biométrico busca en una base de datos una referencia que coincida con la muestra biométrica suministrada y, de encontrarla, devuelve la identidad correspondiente.

Identificación de Grupo Abierto: Tarea biométrica que sigue las condiciones del sistema operativo biométrico más de cerca para determinar si un individuo es parte de la base de datos y encontrar el registro de dicho individuo en la base de datos

Identificación de Grupo Cerrado: Tarea biométrica durante la cual se sabe que un individuo no identificado es parte de la base de datos y el sistema intenta determinar su identidad.

Isla: Cresta diminuta.

L

Lazo: Patrón de huella dactilar en el cual las crestas entran por cualquiera de los dos lados, se curvan acentuadamente y salen cerca del mismo lado por el cual entraron, este patrón presenta un núcleo y un delta.

M

Matching: En ingles, Proceso para coincidencia.

Minucias: Características de las crestas que se utilizan para identificar una imagen de huella dactilar.

Modalidad: Tipo o clase de sistema biométrico.

Monocromático: De un solo color.

Muestra Biométrica: Información o datos computarizados, obtenidos por medio de un dispositivo con sensor biométrico

N

Netbook: Categoría de ordenador portátil de bajo costo y reducidas dimensiones que aporta una mayor movilidad y autonomía

Notebook: En ingles, ordenador portátil.

P

Password: En ingles, contraseña.

Piezoeléctrico: Fenómeno que presentan determinados cristales que al ser sometidos a tensiones mecánicas adquieren una polarización eléctrica en su masa, apareciendo una diferencia de potencial y cargas eléctricas en su superficie.

Piroeléctrico: Fenómeno que presentan determinados cristales que al ser sometidos a cambios de temperatura adquieren una polarización eléctrica en su masa, apareciendo una diferencia de potencial y cargas eléctricas en su superficie.

Pixel: Es el menor elemento de un gráfico que puede recibir un valor de color.

Pixel por Pulgada: Medida de la resolución de una imagen digital. Cuánto más píxeles por pulgada, mayor resolución.

Plantilla: Representación digital de las características distintivas de un individuo, que contiene la información extraída de una muestra biométrica.

Polímero: Compuesto químico, natural o sintético, formado por polimerización y que consiste esencialmente en unidades estructurales repetidas.

Precisión: Término general utilizado para describir cuál es el rendimiento de un sistema biométrico.

Proceso de Autenticación: Los rasgos biométricos se comparan solamente con los de un patrón ya guardado, este proceso se conoce también como uno-para-uno (1:1).

Proceso de Identificación: Los rasgos biométricos se comparan con los de un conjunto de patrones ya guardados, este proceso se conoce también como uno-para-muchos (1:N).

Proceso de Verificación: Proceso de autenticación.

Proceso para Coincidencia: Proceso que incluye la comparación de una muestra biométrica con una plantilla almacenada anteriormente, y el cálculo del grado de semejanza

Profundidad de Color: Se refiere a la cantidad de bits necesarios para representar el color de un píxel en una imagen digital.

Pulling: Técnica donde un proceso verifica repetidamente una condición.

R

Reconocimiento: Distinción de una persona o cosa entre las demás por sus rasgos o características.

Reconocimiento de Huella Dactilar: Modalidad biométrica que utiliza una imagen de huella dactilar de una persona para fines de reconocimiento.

Reconocimiento de Iris: Modalidad biométrica que utiliza una imagen de la estructura física del iris de una persona para fines de reconocimiento.

Reconocimiento de la Geometría de la Mano: Modalidad biométrica que utiliza la estructura física de la palma de la mano de un individuo para fines de reconocimiento.

Reconocimiento de Patrones: Es la ciencia que se ocupa de los procesos sobre ingeniería, computación y matemáticas relacionados con objetos físicos y/o abstractos, con el propósito de extraer información que permita establecer propiedades de entre conjuntos de dichos objetos.

Reconocimiento de Retina: Modalidad biométrica que utiliza una imagen de la estructura física de la retina de una persona para fines de reconocimiento.

Reconocimiento de Rostro: Modalidad biométrica que utiliza una imagen de la estructura física visible del rostro de una persona para fines de reconocimiento.

Registro: Proceso de recopilación de muestra biométrica de un usuario final, conversión de la misma en referencia biométrica y almacenamiento de la referencia en la base de datos del sistema biométrico para posterior comparación.

Rendimiento: Descripción de la medición de las características, como precisión o velocidad, de un sistema biométrico.

Resolución: Cantidad de píxeles por distancia de unidad en la imagen. Describe la definición y claridad de la imagen.

S

Seguridad: Parámetro para determinar lo seguro que resulta un sistema biométrico.

Sensor: Dispositivo formado por células sensibles que detecta variaciones en una magnitud física y las convierte en señales útiles para un sistema de medida o control.

Sensor Biométrico: Dispositivo biométrico que convierte los datos biométricos de entrada en una señal digital, y transmite esta información al dispositivo de procesamiento.

Sistema Biométrico: Es un sistema automatizado capaz de, capturar una muestra biométrica del usuario final, extraer y procesar los datos biométricos de dicha muestra, almacenar la información extraída en una base de datos, comparar los datos biométricos con los datos de una o más referencias y decidir el grado de coincidencia indicando una identificación o verificación de identidad.

Sistema Biométrico Multimodal: Sistema biométrico en el cual dos o más de los componentes de modalidad se utilizan al mismo tiempo.

Sistema Empotrado: Es un sistema informático de uso específico construido dentro de un dispositivo mayor.

SmartCard: En inglés, tarjeta inteligente. Es cualquier tarjeta de tamaño reducido con circuitos integrados que permiten la ejecución de cierta lógica programada.

T

Tasa de Detección e Identificación: Tasa en la cual los individuos que son parte de la base de datos son correctamente identificados en una aplicación de identificación de grupo abierto.

Tasa de Error Igual: Estadística utilizada para mostrar el rendimiento biométrico. Por lo general, la EER es la ubicación del umbral donde la FAR y la FRR son iguales.

Tasa de Fallo de Registro: Incapacidad del sistema biométrico de generar un registro fiable de un usuario.

Tasa de Falso Negativo: Similar FRR, excepto que la FRR incluye la tasa de fallo de registro.

Tasa de Falso Positivo: Porcentaje de veces que un sistema produce una falsa aceptación.

Tasa de Falso Rechazo: Porcentaje de veces que el sistema produce un falso rechazo.

Tasa de Falso Reconocimiento: Similar FAR, excepto que la FAR incluye la tasa de fallo de registro.

Terminación: Minucia al final de una cresta.

U

Umbral: Valor predeterminado de un sistema para las tareas de verificación o identificación. La aceptación o el rechazo de los datos biométricos dependen de si el resultado de coincidencia se encuentra por encima o por debajo del valor.

Usuario: Persona que interactúa con el sistema biométrico.

Usuario Cooperativo: Individuo que intencionadamente proporciona su información biométrica al sistema biométrico para la captura.

Usuario Final: Individuo que interacciona con el sistema para registrarse, ser verificado o identificado.

Usuario No Cooperativo: Individuo que desconoce que su muestra biométrica está siendo recopilada.

V

Valle: El espacio en blanco o bajo relieve entre crestas.

Vulnerabilidad: Potencial para el funcionamiento de un sistema biométrico de verse comprometido por actividad fraudulenta, defecto de diseño, accidente o condición ambiental externa.

W

Wachtdog: Es un mecanismo de seguridad que provoca un reset del sistema en caso de que éste se haya bloqueado.

Acrónimos

ADC: Analog Digital Converter.

AFIS: Automatic Fingerprint Identification System.

AFAS: Automatic Fingerprint Autentication System.

ASCII: American Standard Code for Information Interchange.

CCD: Charge-Coupled Device.

CMOS: Complementary Metal-Oxide-Semiconductor.

Dpi: Dots per Inch.

DSP: Digital Signal Processor.

EER: Equal Error Rate.

FAR: False Acceptance Rate.

FER: Failure-to-Enroll Rate.

FMR: False Match Rate.

FNMR: False NonMatch Rate.

FRR: False Rejection Rate.

FTIR: Frustrated Total Internal Reflection.

FTR: Failure-to-Enroll Rate.

GPIO: General Purpose Input/Output.

IC: Integrated Circuit.

JPEG: Joint Photographic Experts Group.

JTAG: Joint Test Action Group.

Kbps: Kilo Bytes per Second.

Mbps: Mega Bytes per Second.

MISO: Master In Slave Out.

MOSI: Master Out Slave In.

MSB: Most Significant bit.

PC: Personal Computer.

PDA: Personal Digital Assistant.

PIN: Personal Identification Number.

RTFI: Reflection-based Touchless Finger Imaging.

SPI: Serial Port Interface.

SRAM: Static Random Access Memory.

TIC: Tecnologías de la Información y Comunicación.

TTFI: Transmission-based Touchless Fingerprint Imaging.

UART: Universal Asynchronous Receiver-Transmitter.

Índice de Contenidos

Agradecimientos.....	vii
Resumen.....	xi
Glosario.....	xiii
Acrónimos.....	xix
Índice de Contenidos.....	xxi
Índice de Figuras.....	xxiii
Índice de Tablas.....	xxv
Capítulo 1 Introducción y Objetivos.....	1
1.1 Motivación.....	1
1.2 Objetivos.....	1
1.3 Organización de la Memoria.....	2
Capítulo 2 Estado del Arte.....	5
2.1 Biometría.....	5
2.2 Sistemas Biométricos.....	5
2.3 Funcionamiento y Rendimiento.....	5
2.4 Procesos de Autenticación e Identificación Biométrica.....	6
2.5 Modalidades Biométricas.....	7
2.6 Sensores de Huella Dactilar.....	7
2.6.1 Estáticos o Desplazamiento.....	7
2.6.1.1 Estáticos.....	7
2.6.1.2 Desplazamiento.....	8
2.6.2 Tecnologías de Sensores de Huella Dactilar.....	9
2.6.2.1 Ópticos.....	9
2.6.2.1.1 FTIR.....	10
2.6.2.1.2 FTIR con un conjunto de Prismas.....	10
2.6.2.1.3 Fibra Óptica.....	10
2.6.2.1.4 Electro-Óptico.....	11
2.6.2.2 Ultrasónicos.....	12

2.6.2.3 Estado Solido	13
2.6.2.3.1 Campo Eléctrico.....	13
2.6.2.3.2 Presión o Piezoeléctrico	13
2.6.2.3.3 Capacitivos.....	14
2.6.2.3.4 Térmicos	14
2.6.2.4 Sin Contactos	15
2.6.2.4.1 RTFI	16
2.6.2.4.2 TTFI	16
2.6.2.4.3 Touchless 3D	17
2.7 Sistemas de Reconocimiento de Huella Dactilar	18
2.7.1 Registro y Proceso para Coincidencia o Matching	18
2.7.2 Adquisición de Imagen	18
2.7.3 Extracción de Vectores de Características o Patrones.....	18
2.7.4 Proceso de Verificación de Plantilla con Muestra	20
 Capítulo 3 El Sensor de Huella Dactilar AT77C104B	 23
3.1 Justificación	23
3.2 Tecnología del Sensor de Huella Dactilar AT77C104B	23
3.3 Características Técnicas del AT77C104B	24
3.4 Funcionalidad del Sensor	25
3.5 Ventajas del Sensor	29
 Capítulo 4 Plataforma de Desarrollo.....	 31
4.1 MBF310v5 con MCB2100.....	31
4.2 AT77C104B con MCB2100	32
4.3 AT77C104B con LPCBE2000-B	33
 Capítulo 5 Descripción de la Solución	 37
5.1 Consideraciones Generales.....	37
5.2 Solución Software.....	39
 Capítulo 6 Pruebas.....	 47
6.1 Según el número de frames.	47
6.2 Según la profundidad de color.	47
6.3 Usando estabilizador de temperatura.	49
6.4 Otros registros del sensor.	50
6.5 Procesando la imagen.	50
 Capítulo 7 Conclusiones y Líneas Futuras.....	 53
7.1 Conclusiones	53
7.2 Líneas Futuras.....	53
 Bibliografía	 55
 Anexo A Código de la Aplicación.....	 57

Índice de Figuras

FIGURA 2.1: SENSORES DE DESPLAZAMIENTO EN SISTEMAS.	8
FIGURA 2.2: SECUENCIA CAPTURA SENSOR HUELLA DACTILAR POR DESPLAZAMIENTO.	9
FIGURA 2.3: SENSOR FTIR.	10
FIGURA 2.4: SENSOR FTIR CON PRISMA REDUCIDO.	11
FIGURA 2.5: SENSOR FIBRA ÓPTICA.	11
FIGURA 2.6: SENSOR ELECTRO-ÓPTICO.	12
FIGURA 2.7: SENSOR DE ULTRASONIDOS.	12
FIGURA 2.8: SENSOR DE CAMPO ELÉCTRICO.	13
FIGURA 2.9: SENSOR PIEZOELÉCTRICO.	14
FIGURA 2.10: SENSOR CAPACITIVO.	15
FIGURA 2.11: SENSOR TÉRMICO.	16
FIGURA 2.12: SENSOR RTFI.	16
FIGURA 2.13: SENSOR TTFI.	17
FIGURA 2.14: TOUCHLESS 3D.	17
FIGURA 2.15: TIPOS DE MINUCIAS.	19
FIGURA 2.16: SECUENCIA DE EXTRACCIÓN DE MINUCIAS.	20
FIGURA 2.17: UMBRAL DE DECISIÓN AL 50%.	21
FIGURA 3.1: CRONOGRAMA DEL INTERFAZ SPI.	24
FIGURA 3.2: DIAGRAMA DE BLOQUES DEL SENSOR AT77C104B.	25
FIGURA 3.3: PINEADO DEL AT77C104B.	26

FIGURA 3.4: FORMATO DE COMANDO.	28
FIGURA 3.5: FORMATO DE DATOS.	28
FIGURA 4.1: VISTA SUPERIOR MCB2100.	31
FIGURA 4.2: VISTA SUPERIOR EVAL.BOARD AT77C104B.	32
FIGURA 4.3: VISTA SUPERIOR DE LPCEB2000-B SOBRE LPCEB2000-I.	33
FIGURA 4.4: VISTA SUPERIOR INTERFACE.	34
FIGURA 4.5: DIAGRAMA DE BLOQUES DE SISTEMA DESARROLLADO.	35
FIGURA 5.1: DISPOSICIÓN DE PIXELS EN UN FRAME.	38
FIGURA 5.2: CRONOGRAMA RECEPCIÓN DE IMAGEN.	38
FIGURA 5.3: IMAGEN DE HUELLA CAPTURADA.	39
FIGURA 5.4: FUNCIÓN ADQUISICIONDATOS().	42
FIGURA 5.5: FUNCIÓN ORDENA().	43
FIGURA 5.6: DIAGRAMA DE FLUJO DE LA APLICACIÓN.	45
FIGURA 6.1: IMÁGENES A 80, 120 Y 160 FRAMES.	48
FIGURA 6.2: IMÁGENES CON DISTINTOS CONTRASTES.	49
FIGURA 6.3: APLICACIÓN DEL ESTABILIZADOR DE TEMPERATURA EN CONDICIONES AMBIENTALES DESFAVORABLES.	49
FIGURA 6.4: APLICACIÓN DEL ESTABILIZADOR DE TEMPERATURA EN CONDICIONES AMBIENTALES FAVORABLES.	50
FIGURA 6.5: FASES DEL PROCESADO DE LA IMAGEN.	51

Índice de Tablas

TABLA 2.1: COMPARATIVA MODALIDADES BIOMÉTRICAS.	7
TABLA 3.1: DESCRIPCIÓN PINES DEL SENSOR AT77C104B.	26
TABLA 3.2: LISTADO DE LOS REGISTROS DEL AT77C104B.	27
TABLA 4.1: VELOCIDAD DE DESPLAZAMIENTO FRENTE FRECUENCIA DE TRANSMISIÓN.	35
TABLA 5.1: CONTENIDO 4 PRIMERAS DIRECCIONES DE MEMORIA RECEPCIÓN.	42
TABLA 5.2: CONTENIDO 4 PRIMERAS DIRECCIONES DE MEMORIA ORDENADAS.	44

Capítulo 1

Introducción y Objetivos

Este primer capítulo está dedicado a proporcionar una visión global del documento. Primero, se analizan las causas que han llevado a emprender este trabajo. A continuación, se presentan los principales objetivos que se pretenden alcanzar. Por último, se describe la organización y el contenido de esta memoria.

1.1 Motivación

La enorme evolución tecnológica, las redes de ordenadores y tecnologías de la información y comunicación (TICs) de los últimos 20 años ha estimulado la necesidad de una comunicación globalizada e instantánea entre personas físicas o entidades de cualquier índole o magnitud. Se han desplegado infraestructuras que permiten al individuo estar conectado en cualquier lugar y situación, ha permitido el acceso a dispositivos para establecer las conexiones abaratando los productos y fomentado su uso de forma que cualquier tipo de comunicación o conexión se lleva a cabo en cualquier ámbito de forma cotidiana, propiciando el aumento de fraude producido por falsa identidad.

La necesidad de hacer frente a las potenciales situaciones de falsa identidad en este ámbito tecnológico, junto con la evolución antes mencionada, han propiciado la investigación, desarrollo y abaratamiento de tecnologías como los sistemas automáticos de identificación biométrica que hace unos años parecían destinadas únicamente al ámbito forense, militar o gubernamental[1]. Actualmente se dispone de sensores biométricos de reducido tamaño y coste, además son aceptados por los usuarios.

Los sistemas biométricos basados en huellas dactilares, son considerados los sistemas más consolidados, fiables y aceptados en la actualidad,[2] además de ser los de menor tamaño, especialmente si se trata de sensores de huella dactilar por desplazamiento que por sus reducidas dimensiones, parece actualmente la única alternativa de implantación en dispositivos de tamaño reducido como Notebooks, Teléfonos móviles, PDAs, etc.

1.2 Objetivos

El objetivo principal de este Proyecto Fin de Carrera es obtener imágenes de calidad de huellas dactilares a partir de un sensor de adquisición de huella dactilar por desplazamiento.

Para alcanzar este objetivo, ha sido necesario plantear y alcanzar otros objetivos específicos:

Analizar las características del sensor de huella dactilar para su control y adquisición de datos. En este punto el sensor de Fujitsu MBF310v5 también requería plantear y desarrollar un algoritmo para la reconstrucción de la huella dactilar, siendo innecesario este planteamiento con el uso del AT77C104B pues el algoritmo está implementado en su circuitería.

Estudiar el sistema microcontrolador necesario para el desarrollo del sistema completo, y desarrollar la aplicación.

Diseñar y montar las placas de circuito integrado necesarias para hacer el interfaz entre el sistema microcontrolador y el sensor de huella dactilar.

Desarrollar una aplicación que permita la recepción de la imagen desde el sistema desarrollado y la muestre en pantalla.

1.3 Organización de la Memoria

El Contenido de este Proyecto Fin de Carrera esta organizado de la siguiente manera:

- **Capítulo 1: Introducción.**

Se presenta globalmente el Proyecto, explicando la motivación del mismo y presentando los objetivos.

- **Capítulo 2: Estado del Arte.**

En este capítulo se define el concepto de biometría, se muestra una visión general de los sistemas biométricos y se presentan los mas relevantes.

También se introducen las pautas de funcionamiento más comunes, así como los algoritmos más utilizados en la evaluación del sistema y su rendimiento.

A continuación se explican y describen brevemente los procesos de identificación y autenticación.

Asimismo, se incluye un estudio con una descripción de los principales sistemas biométricos.

Por ultimo, el capítulo finaliza con la descripción de las distintas tecnologías existentes en lo referente a sistemas de adquisición de huella dactilar y una explicación mas especifica en lo referente a los procesos de captura, extracción de minucias, registros, patrones y verificación en los sistemas de huella dactilar.

- **Capítulo 3: El Sensor de Huella Dactilar AT77C104B.**

En este capítulo, se presenta en detalle el sensor térmico utilizado en este Proyecto, tras una breve justificación de su elección.

Se detalla su tecnología, características principales y se aborda su modo de funcionamiento.

- **Capítulo 4: Plataforma de Desarrollo.**

Aquí se explican los inconvenientes encontrados y las soluciones empleadas para obtener el hardware final empleado en el sistema de adquisición de huella dactilar.

- **Capítulo 5: Descripción de la Solución.**

Capítulo en el que se explica el desarrollo de la solución del control del sensor, adquisición de datos y su ordenación. Primero se presentan las principales reflexiones ha tener en cuenta en el desarrollo software para posteriormente explicar la secuencia del código.

- **Capítulo 6: Pruebas.**

Capítulo en el que se comparan distintas imágenes adquiridas en función de los parámetros de configuración del sensor, para determinar los más eficientes.

- **Capítulo 7: Conclusiones y Líneas Futuras.**

Este último capítulo sirve para presentar las conclusiones finales consecuencia del trabajo realizado y discutir los resultados obtenidos. Además, se proponen futuras líneas de trabajo.

Capítulo 2

Estado del Arte

En este capítulo se aborda el estudio de los sensores de huella dactilar por desplazamiento, así como de otras tecnologías relacionadas. En primer lugar se hace una introducción la biometría, su funcionamiento generalizado y su evolución. A continuación, se incluye un estudio de los distintos sistemas biométricos. Para terminar, se presentan las distintas soluciones Hardware y Software para el problema que nos ocupa.

2.1 Biometría

Consiste en la identificación de un individuo a través del análisis de sus características físicas o rasgos conductuales. El término se deriva de las palabras griegas "bios" de vida y "métron" de medida.[3]

2.2 Sistemas Biométricos

Son sistemas diseñados para conceder acceso a usuarios basándose en características únicas e inalterables de cada individuo que difícilmente pueden ser duplicadas, perdidas, olvidadas o robadas, pretendiendo reconocer quien es el individuo, contrariamente a los sistemas de acceso mas comunes basados en objetos (llaves, smartcards...) o códigos (passwords, PIN...) que pueden ser perdidos, olvidados, robados o descifrados.[4][5][6][7]

Son precisamente estas características físicas o conductivas de cada individuo lo que ha impulsado el desarrollo de distintos tipos de sistemas para su identificación; desde el punto de vista conductivo podemos analizar la caligrafía o la voz de un individuo, siendo esta última donde mas esfuerzos se han aplicado desde una perspectiva tecnológica. Pero sin duda alguna el desarrollo de los sistemas biométricos, centra sus esfuerzos en la identificación de las características físicas como son, por ejemplo, la Retina, el Iris, la Geometría del la Mano, la Cara o la Huella Dactilar.[5][8]

2.3 Funcionamiento y Rendimiento

La pauta habitual en un Sistema Biométrico, consta de un proceso previo de registro, donde se adquieren uno o varios de los biométricos morfológicos o fisiológicos del usuario, se procesan mediante algún algoritmo numérico para finalmente ser almacenados en una base de datos con la información única para cada individuo. Idealmente, en la autenticación o identificación de un individuo, el sistema adquiere y procesa las características biométricas, las compara con las almacenadas en la base de datos y si concuerda permite el acceso. Las tecnologías actuales tienen tasas de error que varían ampliamente, desde valores bajos como el 60%, hasta altos como el 99,9%.

Alguno de los parámetros definidos para medir el rendimiento de un sistema biométrico son:[5][7][9] Tasa de Falso Positivo (False Acceptance Rate o FAR); es la probabilidad de que un usuario no registrado sea aceptado en cualquier caso. Mide la frecuencia con que un usuario no registrado es admitido por equívoco, actualmente el margen esta entre el 0.0001% y el 0.1%.[8] Y asume intentos pasivos del usuario. A partir

del FAR se puede obtener una medida de la seguridad del sistema de verificación biométrica, según la ecuación, $\text{seguridad} = (1 - \text{FAR})$.

Tasa de Falso Reconocimiento (False Match Rate o FMR). Es la tasa con que los usuarios no registrados son confundidos durante un proceso de comparación de características.

Tasa de Falso Rechazo (False Rejection Rate o FRR); es la probabilidad de que un usuario este registrado pero sea rechazado, Mide la frecuencia con que un usuario registrado es rehusado. Comercialmente su valor varía entre el 0.00066% y el 1%. [8] Conociendo FRR, se puede precisar la conveniencia de un sistema biométrico, de acuerdo con la ecuación, $\text{conveniencia} = (1 - \text{FRR})$. Conforme aumenta FRR menos conveniente resulta el sistema por el alto número acceso denegado de forma errónea.

Tasa de Falso Negativo (False NonMatch Rate o FNMR). Es la tasa con que los usuarios registrados no son identificados durante un proceso de comparación de características.

Es común confundir los conceptos FAR con FMR y FRR con FNMR, sin embargo la principal diferencia consiste en la no contabilización de los rechazos producidos debidos a la baja calidad de la imagen o imposibilidad de adquirirla cuando se trata de FMR respecto a FAR o de FNMR hacia FRR.

Tasa de Fallo de Registro (Failure-to-enroll Rate, FTR o FER); incapacidad del sistema biométrico de generar un registro fiable de un usuario.

Tasa de Error Igual (Equal Error Rate, EER); es la intersección entre FAR y FRR siendo una de las medidas mas usada para determinar el umbral de calidad del sistema biométrico, cuando los errores de aceptación y rechazo son iguales, el umbral se establece en el 50%. Cuanto más bajo es EER el sistema es más fiable. Estos umbrales deben ser ajustados de acuerdo con la garantía requerida.

2.4 Procesos de Autenticación e Identificación Biométrica

En el control de accesos mediante biometría existe una fase de verificación, durante la cual, el dato biométrico o plantilla, ya adquirido por el sistema, es cotejado con el que está siendo sometido a análisis, intentando confirmar la identidad declarada de un individuo, al confrontar la muestra suministrada con una o más plantillas registradas con anterioridad siguiendo uno de los dos procesos fundamentales, el de Autenticación o el de Identificación. [5][7]

El Proceso de Autenticación (o Verificación), también conocido como Uno-Para-Uno (1:1), se basa en la comparación de los rasgos biométricos con los de un patrón ya guardado. El objetivo es confirmar la identidad del individuo.

Los sistemas automáticos para verificación se denominan AFAS.

El proceso de Identificación o Uno-Para-Muchos (1:N), compara los rasgos biométricos con los de un conjunto de plantillas residentes en la base de datos. Pretende identificar al individuo.

Cuando se conoce que el individuo a identificar esta dentro de la base de datos, se dice que la identificación es “de grupo cerrado”.

En identificación “de grupo abierto”, también llamada “lista de vigilancia”, no existe garantía de que el individuo sea parte de la base de datos. El sistema debe determinar si el

individuo es parte de la base de datos y en caso afirmativo proporcionar su identidad (AFIS).

Debido al número de comparaciones y procesamiento que debe llevarse a cabo en el proceso de Identificación, resulta mas rápido el proceso de Autenticación, especialmente conforme va aumentando N en la Identificación.

2.5 Modalidades Biométricas

Lo que sigue a continuación es una comparativa en la que se recogen las diferentes características de los sistemas biométricos más comunes mostrados en la Tabla 2.1.[8]

	Ojo (Iris)	Ojo (Retina)	Huellas Dactilares	Geometría de la Mano	Escritura y Firma	Voz	Cara
Fiabilidad	Muy Alta	Muy Alta	Alta	Alta	Media	Alta	Alta
Facilidad de Uso	Media	Baja	Alta	Alta	Alta	Alta	Alta
Prevención de Ataques	Muy Alta	Muy Alta	Alta	Alta	Media	Media	Media
Aceptación	Media	Baja	Alta	Alta	Muy Alta	Alta	Muy Alta
Estabilidad	Alta	Alta	Alta	Media	Baja	Media	Media
Costo	Muy Alto	Alto	Bajo	Bajo	Alto	Medio	Medio
Incidencias	Enfermedades Oculares, Luz	Enfermedades Oculares, Gafas	Ausencia Miembros	Edad, Ausencia Miembros	Edad, Analfabetismo, cambios	Ruido, Afecciones	Edad, pelo
Tamaño Plantilla (bytes)	512	96	250 - 1000	20	1000 - 3000	10000 - 20000	84 - 1300

Tabla 2.1: Comparativa Modalidades Biométricas.

El Sistema Biométrico, por lo tanto, debe proporcionar el máximo nivel de seguridad, conveniencia y facilidad de uso.

2.6 Sensores de Huella Dactilar

Se puede deducir de la sección anterior, que de entre todos los sistemas biométricos, los sensores de huella dactilar son los que presentan mayor potencial de mercado conjugado con buena aceptación del usuario y un óptimo rendimiento biométrico.

Se trata de un sistema de captura en vivo que puede fabricarse bajo multitud de variantes tecnológicas aunque se pueden agrupar en una de estas familias: Ópticos, Estado Solido, Ultrasónicos y Sin contactos. Independientemente de la familia tecnológica a la que pertenece el sensor, en el mercado existen dos líneas de productos, los sensores de huella dactilar estáticos y los sensores de huella dactilar por desplazamiento.[5][9][10]

2.6.1 Estáticos o Desplazamiento

Mientras que los estáticos llevan mucho tiempo en el mercado, actualmente están surgiendo diferentes tipos de los llamados sensores de huella dactilar por desplazamiento.[10]

2.6.1.1 Estáticos

Hacen uso de una ventana de captura de imagen que tiene el tamaño requerido para la huella dactilar, apoyando el dedo sobre esta ventana durante el tiempo necesario para la captura de la imagen. Este método tiene la ventaja de capturar la imagen en una única

operación. Entre sus desventajas destacan el gran tamaño del dispositivo, incrementando su coste y dificultando la portabilidad, además la posibilidad de que permanezca una huella latente en la ventana de captura la convierte en un poco menos segura.

2.6.1.2 Desplazamiento

Esta tecnología consiste en el uso de una ventana rectangular con una anchura capaz de cubrir el ancho del dedo y unos pocos pixeles de altura, por lo que para obtener la huella dactilar completa, el sensor debe capturar imágenes continuamente mientras el dedo se va desplazando por la superficie.

Están basados en la tecnología del silicio, resultan más baratos de fabricar, requieren poco mantenimiento, prestan gran precisión, un bajo consumo y portabilidad.

Permite reducir el área del dispositivo en un factor de 5, así puede ser usado en un amplio numero de sistemas empotrados, incluyendo sistemas de control de acceso, PCs, PDAs... usándose en cualquier situación o lugar siendo rápido, fiable y preciso, tanto para Identificación como Autenticación.

La Figura 2.1, muestra ejemplos de sensores de huella dactilar por desplazamiento implementados en sistemas.



Figura 2.1: Sensores de Desplazamiento en sistemas.

Puede considerarse como “autolimpiable”, pues el propio desplazamiento del dedo sobre la superficie del sensor imposibilita que permanezcan huellas latentes aumentando la seguridad. Seguridad que se ve reforzada por la dificultad de obtener imágenes fiables habiendo forzado a un individuo a usar el dispositivo, debido a movimientos erráticos o

dedos sudorosos y ante la dificultad de desplazar un dedo artificial con suficiente sensibilidad como para poder reconstruir una huella.

Siempre que el desplazamiento del dedo sobre la superficie del sensor se realice a una velocidad apropiada, el solapamiento de sucesivos fotogramas, debe permitir la reconstrucción de la huella dactilar, a través de la información de la velocidad de desplazamiento proporcionada por el sensor o estimada por la información del solapamiento de capturas consecutivas.

Este sistema, debido a la distorsión generada en la captura por el movimiento del dedo, requiere una corrección de esta distorsión una vez la huella dactilar haya sido reconstruida.

La Figura 2.2 representa el proceso de captura y reconstrucción de huella dactilar.

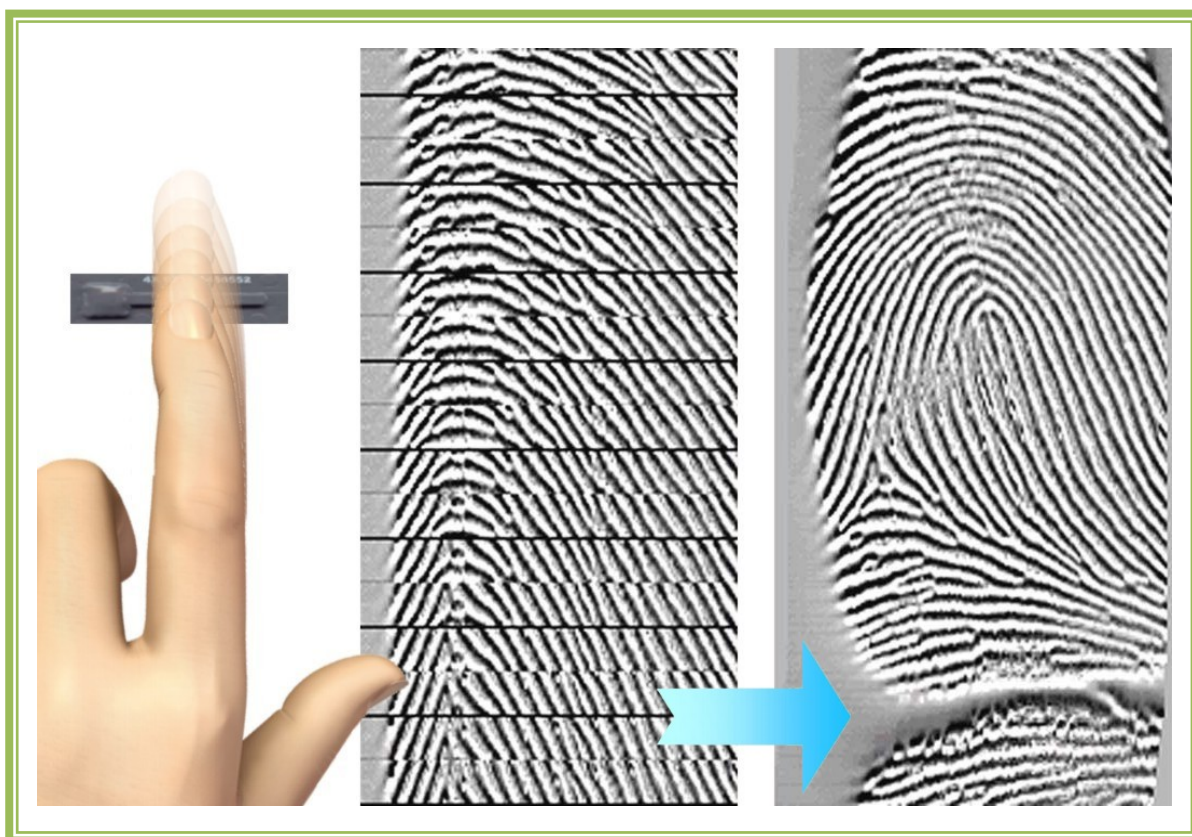


Figura 2.2: Secuencia captura sensor huella dactilar por desplazamiento.

2.6.2 Tecnologías de Sensores de Huella Dactilar

Existen multitud de tecnologías desarrolladas para los sensores de huella dactilar y otras están aun en desarrollo. De todas ellas, quizás las mas demandadas sean las ópticas y las de estado solido; capacitiva, térmica... en esta sección se introducirá cada una de ellas y se valorará cual presenta mayores prestaciones.

2.6.2.1 Ópticos

Se tratan de una variante de cámara digital que captura la imagen de la huella dactilar cuando la yema del dedo se apoya en una ventana de cristal, debidamente iluminada. Los principales inconvenientes de esta tecnología son: las huellas latentes en el cristal que se van acumulando en sucesivas lecturas y que puede ser engañado con imitaciones de huellas.

Los sistemas de captura ópticos se desarrollan con distintas tecnologías que se presentan a continuación.

2.6.2.1.1 FTIR

Es la más antigua y usada de las técnicas de adquisición de huella dactilar en la actualidad.

Consiste en la adquisición de la huella a partir de la luz reflejada a través de un prisma piramidal invertido, el dedo se apoya en un cristal situado en la base del prisma, produciéndose un contacto directo entre las crestas y el cristal y un espacio entre los valles y el cristal; desde un lateral del prisma se aplica una luz hacia la base del mismo que será totalmente reflejada en las zonas de valle y ligeramente absorbida en las zonas de cresta. Obteniéndose así la imagen de la huella en la otra cara del prisma y capturada por las cámaras.

Entre los inconvenientes de esta técnica está la distorsión trapezoidal sobre la imagen que debe ser corregida.

La Figura 2.3 muestra un esquema de este tipo de sensores.

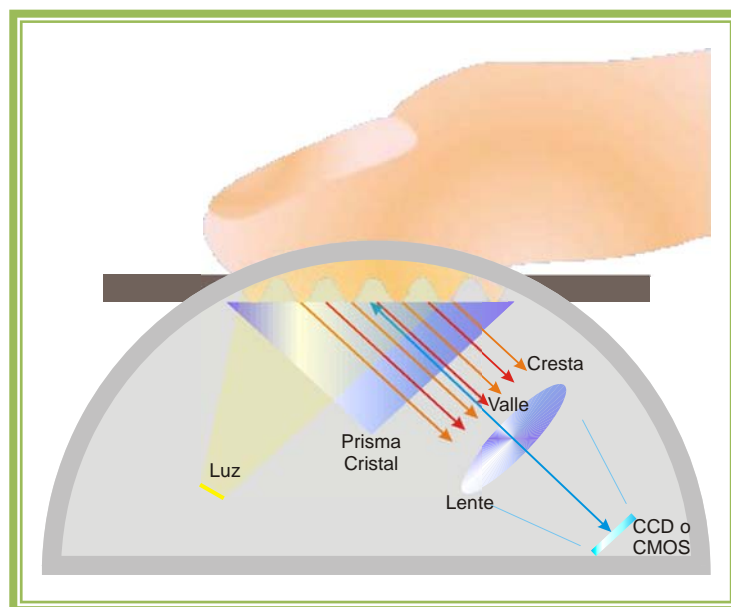


Figura 2.3: Sensor FTIR.

2.6.2.1.2 FTIR con un conjunto de Prismas

Es una alternativa para reducir el tamaño de los FTIR, consiste en sustituir el prisma por un conjunto de pequeños prismas alineados y aunque el objetivo de reducción del tamaño se alcanza, empeora la distorsión de la imagen adquirida. Un esquema de esta técnica se muestra en la Figura 2.4.

2.6.2.1.3 Fibra Óptica

Una mayor reducción del tamaño en los sensores de huella dactilar ópticos se consigue al sustituir el prisma y las lentes por una pletina de fibra óptica. El dedo entra en contacto directo con una lado de la pletina de fibra, mientras que el otro está unido al receptor óptico, bien CCD o CMOS.

Aunque se reduce el tamaño del dispositivo sustancialmente, como se puede apreciar en la Figura 2.5, aun resulta muy caro.

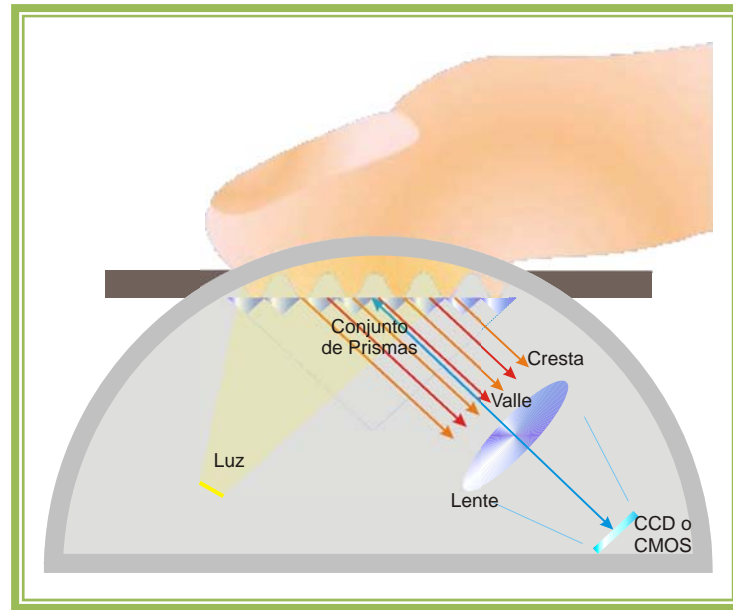


Figura 2.4: Sensor FTIR con Prisma Reducido.

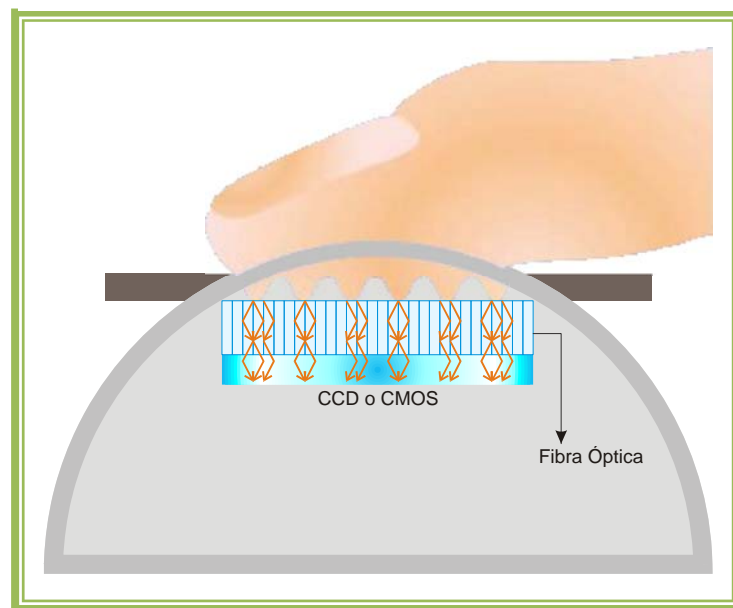


Figura 2.5: Sensor Fibra Óptica.

2.6.2.1.4 Electro-Óptico

Estos sensores constan de dos capas principales; la primera de ellas esta formada por un polímero capaz de emitir luz en función del voltaje aplicado en una de sus caras. Una vez más como las crestas tocan el polímero y los valles no, existe una diferencia de potencial entre unas zonas y otras y la cantidad de luz emitida varia. La segunda capa que esta unida a la primera, esta constituida por un array de fotodiodos encargados de transformar la luz recibida en la imagen digital. A pesar de la gran reducción del tamaño, la calidad de la imagen de estos sensores no es comparable a la obtenida con los FTIR.

La Figura 2.6 muestra un esquema de esta tecnología.

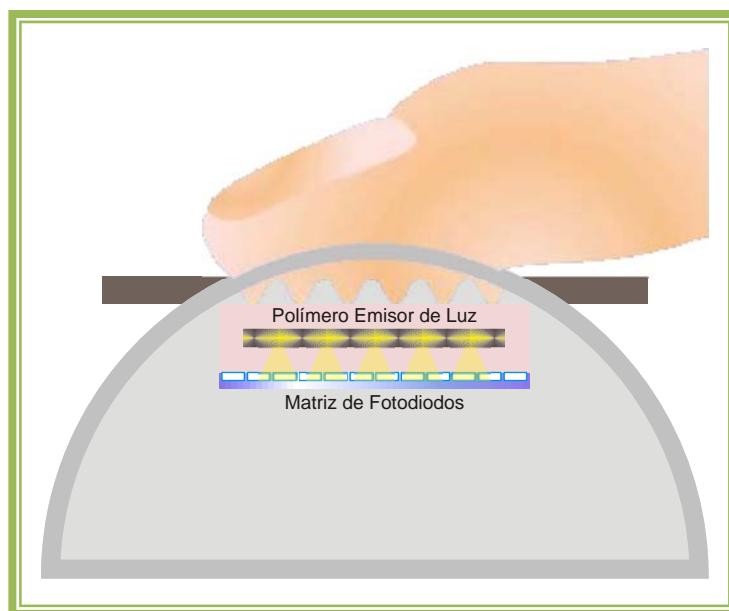


Figura 2.6: Sensor Electro-Óptico.

2.6.2.2 Ultrasonicos

La adquisición por ultrasonidos, se puede ver como una especie de ecografía. Se basa en el envío de señales acústicas hacia la yema del dedo y capturar el eco para tras su procesado poder representar la huella dactilar.

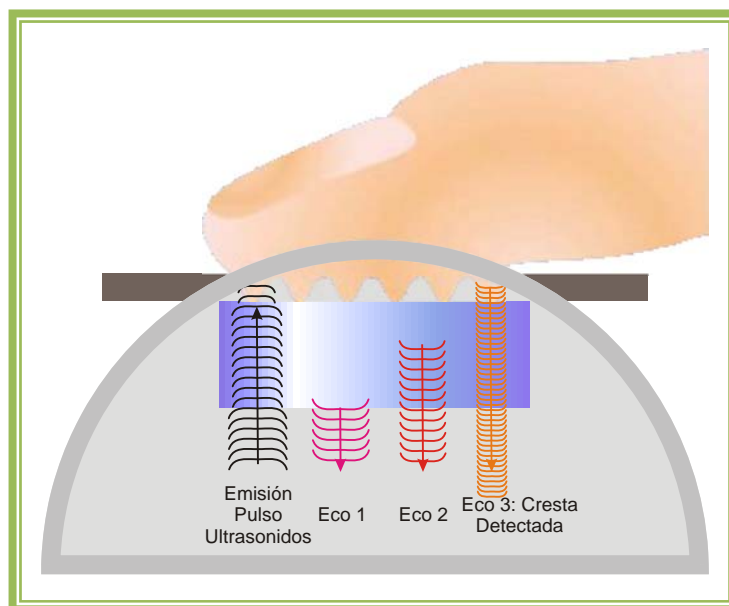


Figura 2.7: Sensor de Ultrasonidos.

El dispositivo consta de un transmisor, que genera pulsos acústicos, y el receptor, que detecta la respuesta cuando los pulsos transmitidos rebotan en la superficie del dedo de distinta manera si inciden sobre un valle o una cresta, como se muestra en la Figura 2.7.

Esta técnica se muestra robusta frente a la suciedad y grasas acumuladas en el dedo, además de obtener imágenes de buena calidad. El inconveniente es que todavía es cara y no está desarrollada lo suficiente.

2.6.2.3 Estado Solido

También conocidos como sensores de silicio, surgieron para abaratar los dispositivos y reducir el tamaño de los mismos. Todos los sensores basados en silicio constan de una matriz de píxeles, siendo cada píxel un pequeño sensor.

Las principales tecnologías desarrolladas para la adquisición de la imagen digital son: Campo Eléctrico, Piezoeléctrico, Capacitivo y Térmico.

2.6.2.3.1 Campo Eléctrico

Consideran la carga con una onda de radio de baja intensidad en el dedo como un transmisor, y las variaciones entre valles y crestas como una antena de array debidamente sintonizada. Para conseguirlo, el sensor consta de un anillo conductor que genera una señal sinusoidal y una matriz de antenas activas que reciben la señal generada por el anillo y modulada por la morfología de la piel. Es precisamente la dependencia de las propiedades fisiológicas de la piel lo que le hace ser un sensor difícil de engañar con dedos artificiales.

Un ejemplo de este tipo de sensores se muestra en la Figura 2.8.

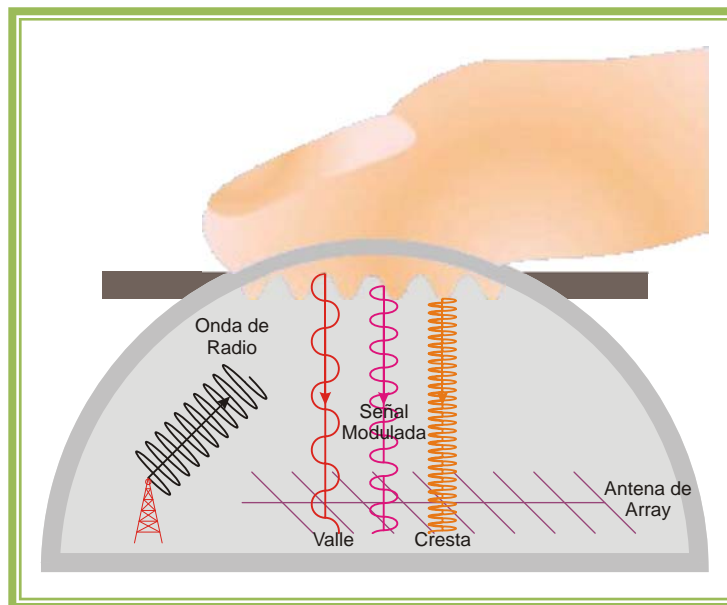


Figura 2.8: Sensor de Campo Eléctrico.

Sin embargo, la necesidad del contacto directo entre el anillo generador y el dedo puede causar rechazo del usuario por la alta temperatura que puede alcanzar la superficie de contacto.

2.6.2.3.2 Presión o Piezoeléctrico

Los sensores piezoeléctricos han sido diseñados para generar una señal eléctrica cuando son sometidos a una presión, esto es el denominado efecto piezoeléctrico. De tal forma que a más presión mayor corriente se genera. Una vez más, la diferencia de

distancias existentes entre las crestas y los valles con respecto a la superficie del sensor, se traducen en una diferencia de presión y por tanto en una diferencia de corriente.

Desafortunadamente, estos sensores no son lo suficientemente sensibles como para detectar diferencias de presión tan pequeñas y se pueden engañar fácilmente.

La Figura 2.9 muestra una representación de este tipo de sensores.

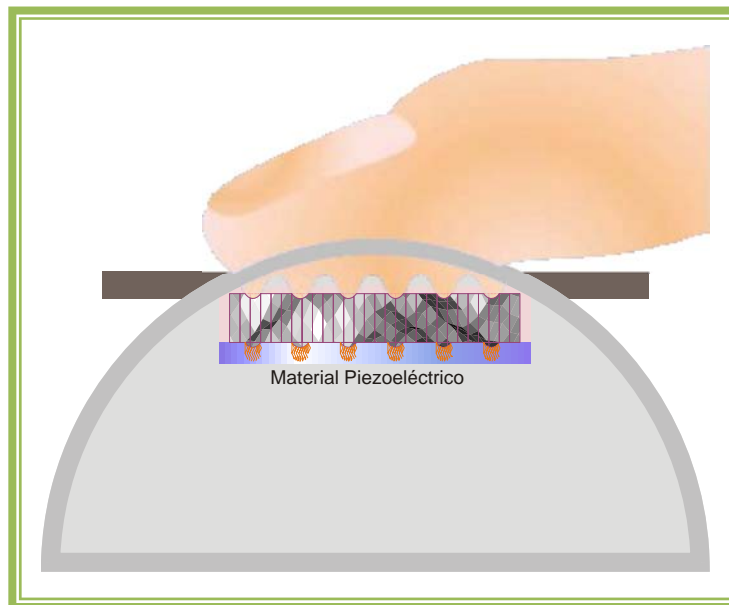


Figura 2.9: Sensor Piezoeléctrico.

2.6.2.3.3 Capacitivos

Es una de las tecnologías mas usada de entre los sensores de silicio. El sensor formado por un array bidimensional de pixeles sensibles a la carga electrostática, se puede considerar como una de las pletinas del condensador, mientras que la otra pletina la constituiría el dedo creándose pequeñas cargas eléctricas entre las dos pletinas. Las variaciones en el dieléctrico entre una cresta (formada principalmente por agua) y un valle (compuesto por aire) causan una variación capacitiva local. Esto permite que las crestas y los valles sean identificados, y la imagen sea construida.

En ocasiones los dedos demasiado húmedos, por el sudor o incluso demasiado secos, dificultan la adquisición de la huella dactilar, una ventaja de los sensores capacitivos, es que permite ajustar parámetros para mejorar la lectura en estas condiciones; desafortunadamente también tiene inconvenientes, como la necesidad de limpieza frecuente para evitar grasas o suciedad que perjudique la calidad de la imagen, o la posibilidad de dañar el dispositivo con descargas electrostáticas residuales generadas por el dedo.

La Figura 2.10 muestra una representación de los sensores capacitivos.

2.6.2.3.4 Térmicos

Un material piroeléctrico es aquel capaz e convertir una diferencia de temperatura en una tensión. Un sensor de huella dactilar basado en este material, mide la diferencia de

temperatura entre los sensores (píxeles) que están en contacto con las crestas y los que están bajo los valles que no están en contacto directo.

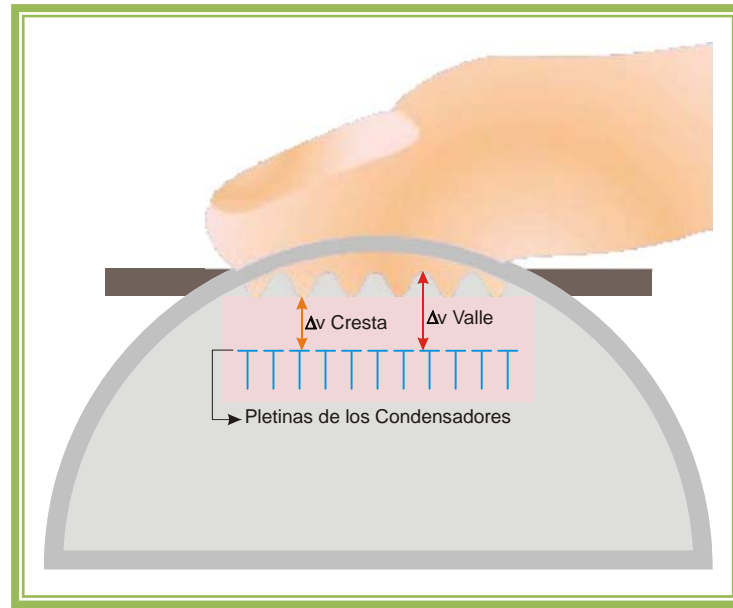


Figura 2.10: Sensor Capacitivo.

El enfoque térmico no se ve afectado por condiciones ambientales extremas, es robusto frente a descargas electrostáticas y ante el fraude con dedos artificiales. Sin embargo, al alcanzarse rápidamente el equilibrio térmico, la diferencia de temperatura inicial que generaba la imagen desaparece y así mismo la imagen. Esto hace que esta técnica se emplee principalmente en los sensores de huella dactilar por desplazamiento, pues al ir desplazando el dedo por la superficie del sensor se crea una ruptura constante del equilibrio térmico.

La representación de un sensor térmico se muestra en la Figura 2.11.

2.6.2.4 Sin Contactos

Cuando un dedo toca o se desliza sobre una superficie, la piel se deforma. La presión ejercida, el estado de la piel y la proyección de una superficie tridimensional irregular (dedo) sobre un plano bidimensional, introducen distorsión, ruido e inconsistencias en la huella capturada. Para tratar de solucionar estos problemas está surgiendo un nuevo enfoque en lo referente a tecnologías de captura de huella dactilar, los sistemas de huella dactilar Sin Contactos o Touchless. Debido a la falta de contacto entre el dedo y una superficie rígida, no produce ningún tipo de deformación en la piel durante la captura.

Los enfoques utilizados en la captura de huellas digitales se pueden agrupar en dos familias: Imágenes basadas en la Reflexión Sin Contactos (RTFI, Reflection-based Touchless Finger Imaging) y las Imágenes Basadas en la Transmisión Sin Contactos (TTFI, Transmission-based Touchless Fingerprint Imaging).[9]

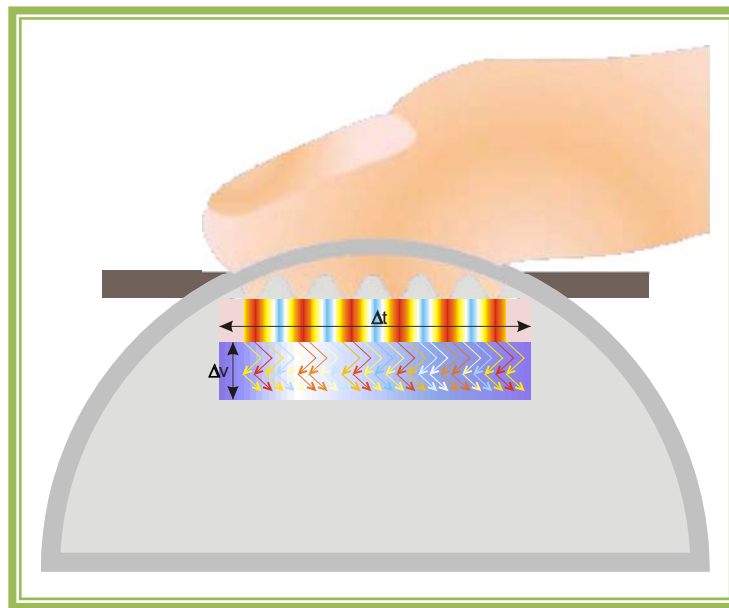


Figura 2.11: Sensor Térmico.

2.6.2.4.1 RTFI

Se emite luz monocromática sobre el dedo y la que es reflejada se captura a través de la cámara CCD o CMOS, según se indica en el esquema de la Figura 2.12.

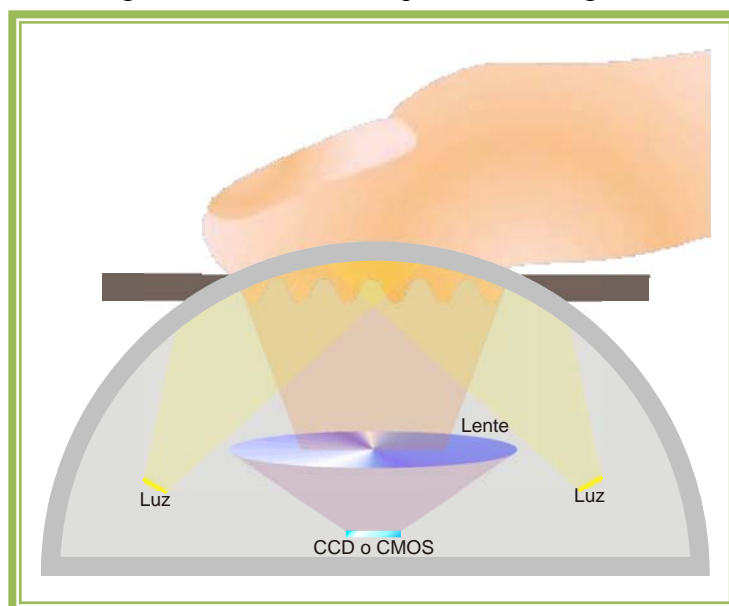


Figura 2.12: Sensor RTFI.

2.6.2.4.2 TTFI

La luz que atraviesa el dedo, normalmente luz natural o ambiente, es capturada en la cámara CCD o CMOS, como muestra la Figura 2.13.

Hay que decir que en cualquiera de los dos casos, la imagen resultante es de menor contraste que las obtenidas mediante la tecnología FTIR, lo que es importante para una buena extracción de minucias y de esta forma, lo que parecía ser una ventaja al evitar la

deformación de las huellas se convierte en un pequeño inconveniente. Además si la superficie del dedo no es iluminado perpendicularmente se producen sombras en los valles debido a las crestas que se traducen en una mala representación de los detalles pequeños para lo que se debe recurrir a técnicas de iluminación mas sofisticadas para evitar estos errores de representación y obtener imágenes mas contrastadas, aumentando el tamaño del dispositivo y su coste.

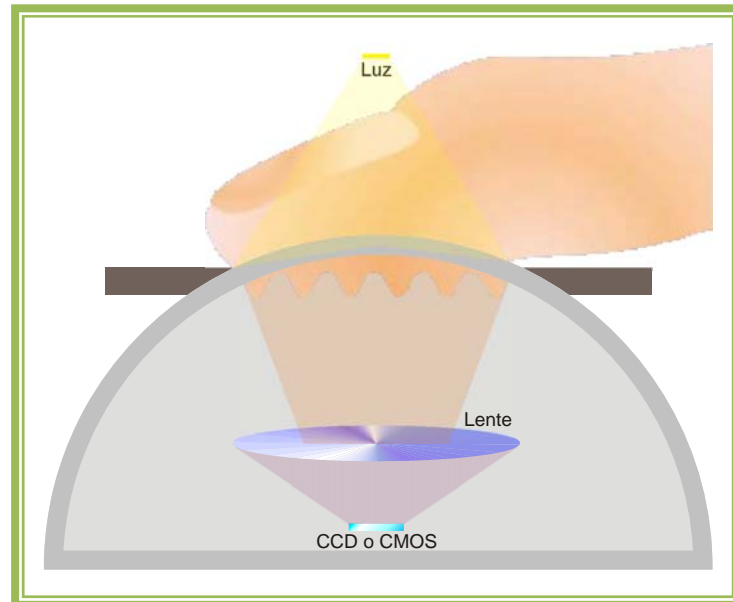


Figura 2.13: Sensor TTFI.

2.6.2.4.3 Touchless 3D

Basadas en el enfoque Touchless, se toman muestra del dedo desde diferentes puntos de vista, por medio de un sistema multicámara o la combinación de cámara y espejos.

Esta nueva tecnología de muestreo representada en la Figura 2.14, resuelve algunos de los problemas asociados a los sensores de contacto como son los debidos a una mala posición del dedo, deformación de la piel, suciedad o huellas latentes, además también aborda algunas de las debilidades de los Touchless vistos anteriormente.

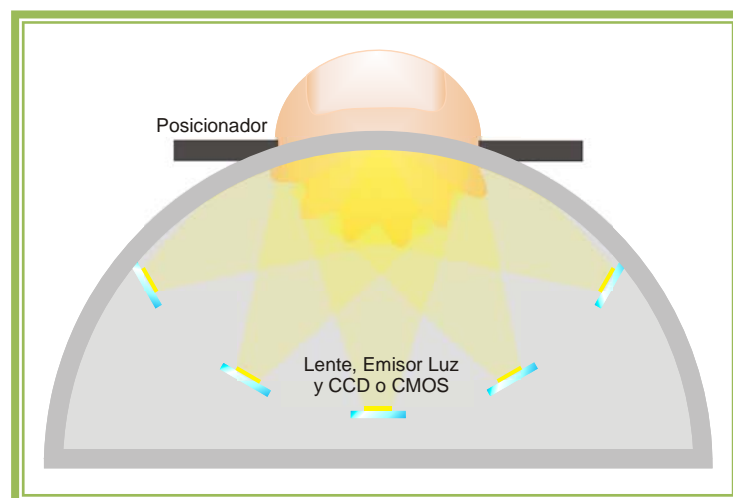


Figura 2.14: Touchless 3D.

Sin embargo, esta tecnología presenta nuevos desafíos. Por ejemplo, la compatibilidad entre las imágenes 3D obtenidas, pues proporcionan un nuevo patrón de minucias, y las 2D disponibles actualmente.

2.7 Sistemas de Reconocimiento de Huella Dactilar

Como se ha hecho referencia anteriormente, el reconocimiento de huella dactilar se utiliza en multitud de aplicaciones y dispositivos, pero independientemente de la tecnología utilizada por los sensores o el tipo de sensor y del software usado para la adquisición de la plantilla, la extracción y la comparación, todas tienen en común una serie de técnicas.[5][9][10]

2.7.1 Registro y Proceso para Coincidencia o Matching

En todo sistema biométrico basado en sensores de huella dactilar, debe haber una primera fase preparatoria, que consiste en el registro en el sistema de las plantillas de toda la población identificable.

Este registro consiste en la captura y almacenamiento en una base de datos de la plantilla asociada a un individuo, esta plantilla puede ser la imagen de la huella dactilar o un conjunto de datos únicos y representativos de la misma conocidos como minucias. Resulta evidente pensar que este registro se debe llevar a cabo en condiciones seguras.

Cuando el sistema esta operativo, las huellas se capturan y se procesan de la misma forma en se hizo en el registro, el dato que se obtiene, denominado muestra, se coteja con las plantillas de la base de datos de la población registrada, o un subconjunto. Si existe alguna coincidencia se da por terminado el proceso de Identificación y se concede el acceso. Cuando el subconjunto de comparación es de 1 individuo se trata de un proceso de Autenticación. Este proceso genérico es el que se conoce como Coincidencia o Matching.

2.7.2 Adquisición de Imagen

Consiste en hacer uso de alguno de los distintos dispositivos presentados en el capítulo anterior para disponer de una imagen de mapa de bits, con una resolución suficiente de toda o parte de la huella dactilar.

2.7.3 Extracción de Vectores de Características o Patrones

Por razones de seguridad, así como de capacidad de almacenamiento, no es prudente almacenar las imágenes completas de las huellas dactilares en el sistema de reconocimiento de huella dactilar. Lo habitual es hacer un backup de la imagen completa durante el registro en un lugar seguro, pero para el funcionamiento habitual del sistema lo que se hace es, extraer una plantilla única de la imagen en función del método de reconocimiento de patrones o el principio de minucias.

Las huellas digitales no solo se componen de segmentos de curva continuos en forma de arcos, curvas y espirales, sino que también cambian de dirección se cortan o bifurcan, los puntos donde estos cambios ocurren se denominan minucias, a continuación se detallan los tipos de minucias y se muestran una representación de ellas en la Figura 2.15.[1]

- **Laguna:** Cresta que se divide y vuelve a unirse formando una superficie cerrada.
- **Isla:** Cresta diminuta, prácticamente un punto.

- **Bifurcación:** Punto donde la cresta se divide en dos ramas.
- **Espolón:** Cresta que se bifurca y una de las ramas tiene una longitud muy pequeña.
- **Terminación:** Donde la cresta termina abruptamente.
- **Cruce:** Cresta que une otras dos crestas paralelas.

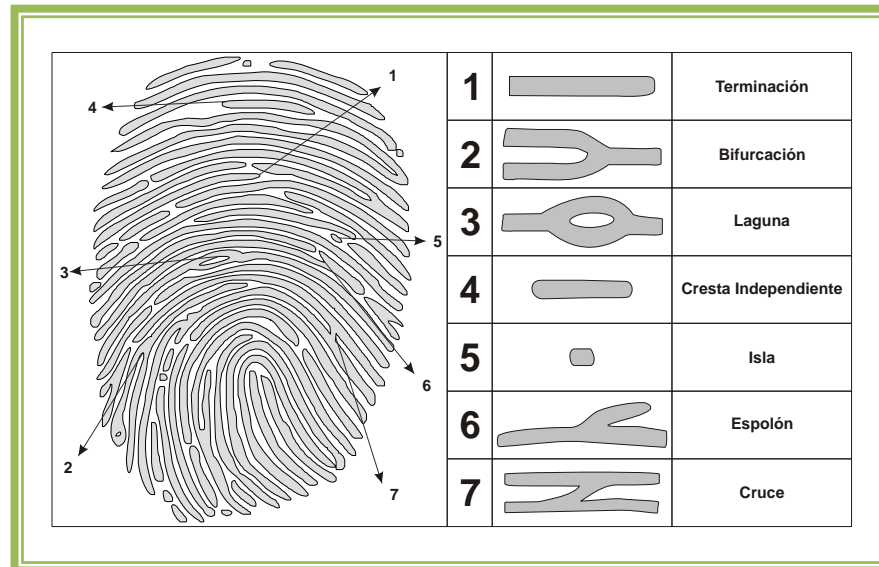


Figura 2.15: Tipos de Minucias.

Las minucias pueden pertenecer a dos categorías:

- **Básicas:** Son las minucias de terminación y bifurcación.
- **Compuestas:** El resto de minucias pertenecen a esta categoría, pues se originan a partir de las básicas.

Y los principales atributos usados para su almacenamiento son:

- **Dirección:** La determina la dirección de la cresta a la que pertenece.
- **Posición:** Posición geométrica de la minucia respecto a un eje de referencia.
- **Frecuencia Espacial:** Inversa de la distancia entre dos crestas consecutivas.
- **Curvatura:** Índice de variación en la dirección de las crestas.

Durante el registro, este proceso proporciona la Plantilla de Registro, y durante la verificación la Plantilla de Muestra, siendo exactamente el mismo proceso en ambos casos, que se muestra generalizado en la Figura 2.16.

Algunas de las razones para utilizar este procedimiento:

- En cada huella dactilar existen entre 30 y 40 minucias, típicamente 36, si cada una se puede almacenar en 4 bytes, toda la información necesaria para la verificación de un individuo a partir de su huella dactilar se puede almacenar en tan solo 144 bytes, en lugar de los 140 Kbytes aproximados de la imagen completa.
- La Huella Dactilar, no se puede recuperar desde la plantilla. de esta forma se reducen las posibilidades de fraude, en caso de poder acceder a la base de datos.

- La plantilla puede ser comprimida con cualquier algoritmo de compresión de datos, e incluso someterse a un proceso de codificación, muy útil en aplicaciones que requieran SmartCards, donde la capacidad de almacenamiento es baja y requieren mucha seguridad.

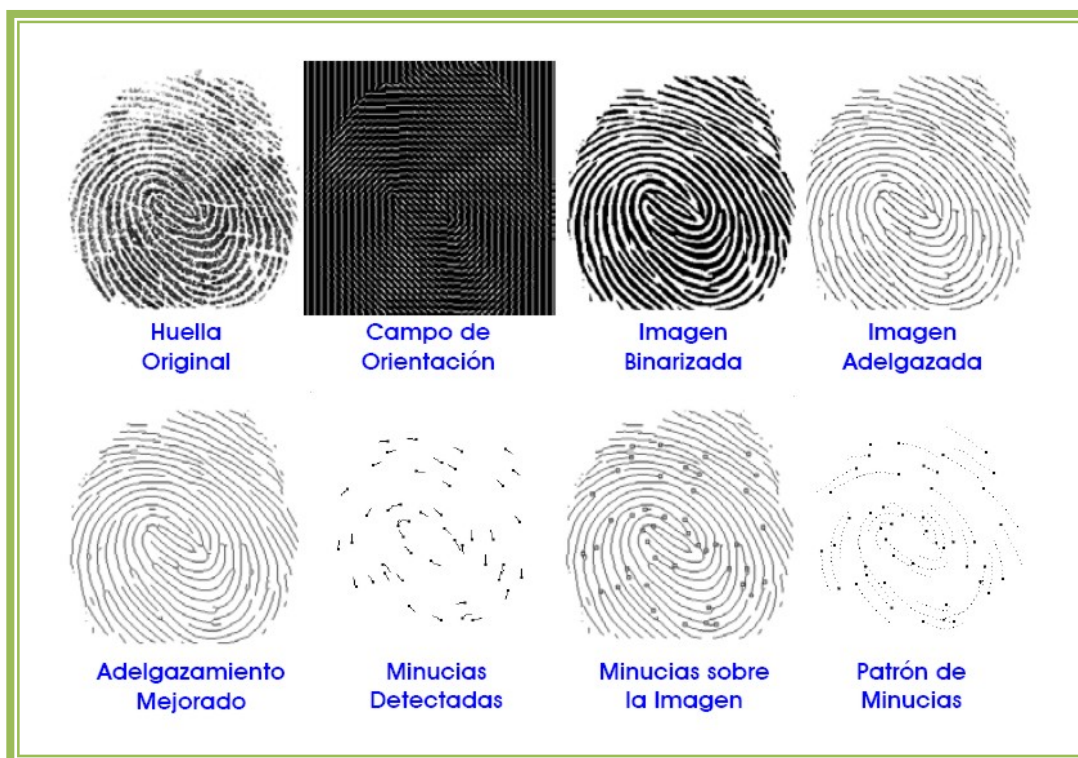


Figura 2.16: Secuencia de extracción de minucias.

Habitualmente la extracción de la plantilla la lleva a cabo un software de terceros, que cumple con los estándares industriales.

2.7.4 Proceso de Verificación de Plantilla con Muestra

El ultimo paso en el proceso de verificación es la comparación de la plantilla muestreada con un conjunto de plantillas registradas en el caso de Identificación, o con una única plantilla registrada en el caso de de Autentificación.

Resulta prácticamente imposible que la imagen muestreada sea exactamente idéntica a la registrada bit a bit. Esto se debe a los errores de escaneo, diferencias en la alineación de la imagen y los errores introducidos en la extracción de minucias. Por tanto, se requiere de un algoritmo de coincidencia que evalúe distintas orientaciones de la imagen y el grado de correspondencia de las minucias asignándole un valor a la coincidencia. Por lo que es necesario establecer un umbral de decisión.

Este umbral de decisión lo determina el EER, teniendo en cuenta que todos los sistemas de reconocimiento de huella dactilar pretenden minimizar FAR y FRR, pero que existe un compromiso entre ellos, tal que reduciendo FAR incrementa FRR y viceversa. El umbral necesita ser ajustado, para minimizar las consecuencias de los dos tipos de error, como se indica en la Figura 2.17.

En la mayoría de las situaciones, FAR es más decisivo, pues se permite el acceso a impostores con todo lo que ello conlleva.

En ocasiones una imagen de alta calidad es esencial para mantener FAR y FRR al mínimo.

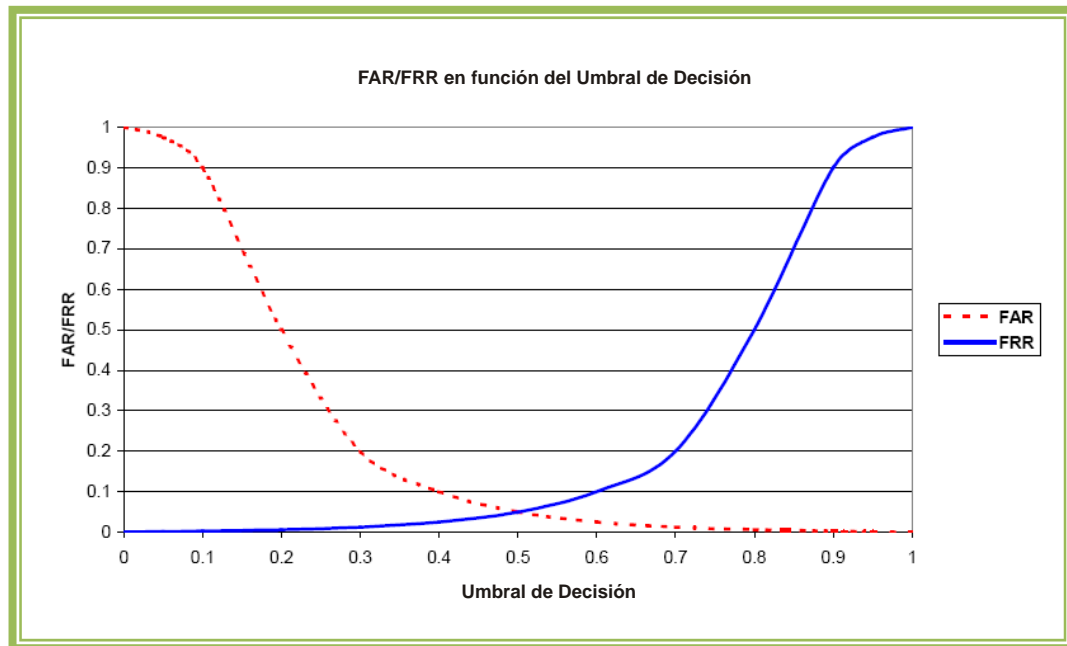


Figura 2.17: Umbral de decisión al 50%.

Capítulo 3

El Sensor de Huella Dactilar AT77C104B

3.1 Justificación

Como ya se ha comentado en capítulos anteriores, la necesidad de sistemas de seguridad eficaces cada vez se hace más evidente, especialmente cuando se trata de proteger sistemas informáticos, aplicaciones de Internet, teléfonos móviles y en general cualquier otro dispositivo o servicio frente a accesos no autorizados o su utilización.

Los sistemas biométricos tratan de identificar individuos mediante la medición de determinadas características físicas o fisiológicas para garantizar accesos seguros a todos estos dispositivos o servicios. Como ya hemos visto, de entre los distintos tipos de sistemas biométricos, los sensores de huella dactilar son los más aceptados, no solo por sus bajos costes, reducidos tamaños y fiabilidad, sino también por su facilidad de uso y predisposición de los usuarios.

A lo largo de este capítulo se presentara el Sensor de Huella Dactilar Térmico por Desplazamiento diseñado por Atmel, AT77C104B empleado en este estudio junto con un Microprocesador de NXP, el LPC2292 para desarrollar un sistema de adquisición de huella dactilar por desplazamiento.

3.2 Tecnología del Sensor de Huella Dactilar AT77C104B

El sensor de huella dactilar AT77C104B fabricado por Atmel,[11] esta basado en la tecnología de sensores térmicos de desplazamiento haciendo uso de material piroeléctrico, que es aquel capaz de generar una corriente, a partir de un diferencial de temperatura. Esta propiedad permite que las crestas de la huella dactilar, que están en contacto directo con la superficie del sensor, produzcan una diferencia de temperatura distinta a la generada por los valles, que no lo están. Este diferencial de temperatura entre crestas y valles se traducen en diferenciales de corriente, debido al material piroeléctrico, que a su vez producirán imágenes mas o menos contrastadas, sin embargo este diferencial de temperatura pronto desaparece debido al equilibrio de temperatura alcanzado entre el sensor y el dedo, estabilizándose la temperatura del pixel.

La ruptura del equilibrio térmico, se consigue con el desplazamiento del dedo sobre la superficie del sensor y así permitir adquirir una huella dactilar estable. La captura de la imagen de la huella dactilar se desarrolla mientras el dedo se va desplazando por el dispositivo verticalmente.

El AT77C104B combina la capacidad de capturar y convertir la imagen resultante de desplazar el dedo por la superficie del sensor, esto es, que dispone del hardware necesario para reconstruir la imagen capturada según se va adquiriendo, reduciendo el coste, consumo y aumentando la velocidad operacional.

Se puede concluir que las ventajas de este sensor resultan de la integración en un único dispositivo de la tecnología de desarrollo; sensores térmicos, el método de desplazamiento; permitiendo romper el equilibrio térmico y la capacidad de reconstruir la huella mientras se está capturando; además de funcionalidades adicionales como la

posibilidad de uso a modo de ratón y la certeza de que la superficie del sensor permanece limpia, lo que implica la no existencia de huellas latentes aumentando su seguridad.

3.3 Características Técnicas del AT77C104B

El AT77C104B trabaja con tensiones comprendidas entre los 2,3 y los 3,6 V, en un rango de temperaturas de entre $-40\text{ }^{\circ}\text{C}$ y $+85\text{ }^{\circ}\text{C}$ y con un consumo máximo aproximado de 6 mA cuando se esta adquiriendo una imagen y de menos de 10 μA en *Sleep mode*.

Se compone de una matriz de 8 filas por 232 columnas, proporcionando un total de 1856 pixeles dependientes de la temperatura por cada frame leído. Añade una columna adicional con un patrón específico utilizado para calibración e identificación de frames. El tamaño de pixel de $50 \times 50\text{ }\mu\text{m}$ resulta en una resolución de 500 dpi en una superficie de imagen de $0,4 \times 11,6\text{ mm}$. Lo que se puede interpretar como una muy buena resolución, pues cuanto mayor resolución, habrá mayor diferenciación entre crestas y valles y por tanto mejor detección de las minucias.

El rango dinámico o profundidad de color, es de 16 tonos en la escala de grises, resultando de la codificación de la intensidad de cada pixel con 4 bits. Para la cuantificación de los pixeles dispone de un convertidor analógico/digital de 8 bits.

Cuando se configura para adquirir huellas digitales, el sensor se muestra totalmente pasivo, utilizando la energía térmica que radia el dedo sobre el sensor para obtener las medidas. Ocasionalmente, en condiciones tales que la temperatura ambiente y corporal difiere en menos de 1 grado se puede activar el estabilizador de temperatura incrementando suavemente la temperatura del sensor y así, proporcionando más contraste a la imagen. Con objeto de poder controlar el consumo por el uso del estabilizador de temperatura, el dispositivo lleva implementado un watchdog, capaz de detener el calentamiento del sensor tras un tiempo determinado y comprendido entre un mínimo de 2 Sg y un máximo de 7 Sg y el nivel de potencia empleado para calentar el sensor es programable, pero existen dos valores por defecto que son de 50 mW y 100 mW.

Dispone de 2 interfaces SPI, uno para el interfaz de control, encargado de las comunicaciones y configuración, (Slow SPI) con reloj programable hasta los 200 Kbps, habilitado por la señal /SSS, y el otro para la transferencia de imagen (Fast SPI), controlado con la señal /FSS y reloj programable hasta los 16 Mbps, proporcionando una tasa máxima de 2146 frames por segundo, permitiendo velocidades de desplazamiento del dedo de hasta 48 cm/s. Finalmente la reconstrucción de la imagen capturada se procesa internamente en el sensor, soportando velocidades de desplazamiento del dedo comprendidas entre los 2 y los 20 cm/Sg.

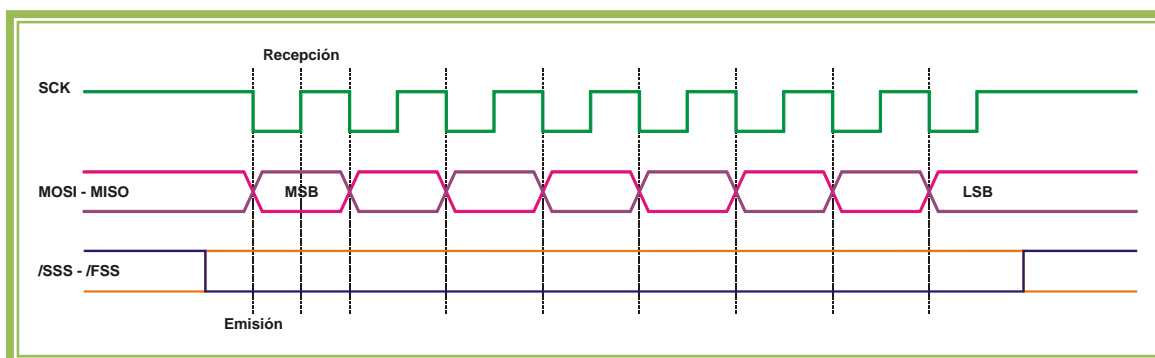


Figura 3.1: Cronograma del interfaz SPI.

El AT77C104B siempre opera en modo esclavo dependiendo del host (Microcontrolador, DSP,...) que actúa como maestro proporcionando el reloj del interfaz, *SCK*, soportando una única configuración de polaridad y fase, mostrada en la Figura 3.1:

- El tiempo de inactividad es en el nivel alto, Polaridad = 1.
- Los datos son enviados por el sensor en el flanco de bajada, y recibidos en el flanco de subida, Fase = 0.

Siempre que las señales de selección del SPI (*/SSS* o */FSS*), estén a nivel bajo según corresponda una u otra.

3.4 Funcionalidad del Sensor

El sensor y toda la circuitería de reconstrucción de datos han sido implementados en un dispositivo de 1,5 x 15 mm. En la Figura 3.2 se presenta un diagrama de bloques de esta circuitería donde se distinguen la matriz de 232 x 8 pixeles, el convertidor analógico-digital, el oscilador interno, los registros de control y estado, y los dos bloques que componen los interfaces SPI, tanto Slow SPI como Fast SPI.

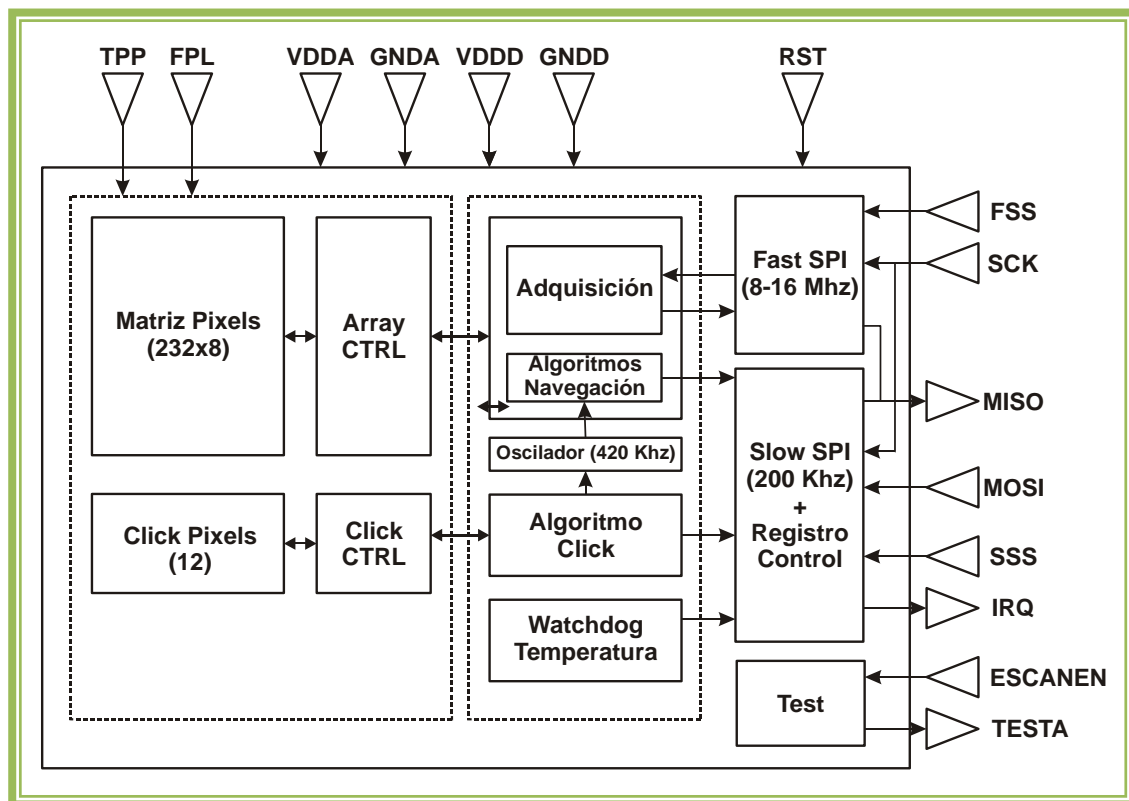


Figura 3.2: Diagrama de bloques del sensor AT77C104B.

En la Tabla 3.1, se identifican cada uno de los pines del dispositivo y se muestra una breve descripción de su función.

Se debe tener la precaución de no activar */SSS* y */FSS* a la vez, si las dos líneas se encuentran a '0' lógico al mismo tiempo, el sensor se configura en modo *SCAN TEST*.

La Figura 3.3, muestra la vista posterior del dispositivo y la disposición de los pines.

Numero de Pin	Nombre	Tipo	Descripción
1	NC		
2	NC		
3	NC		
4	NC		
5	GNDD	G	Masa Digital
6	GNDA	G	Masa Analógica, conectada a GNDD
7	VDDD	P	Alimentación Digital
8	VDDA	P	Alimentación Analógica, conectada a VDD
9	SCK	I	Reloj del Interfaz Serie (SPI), común a SSPI y FSPI
10	TESTA	IO	Reservado para Test Analógicos, NC
11	MOSI	I	Línea Recepción Datos del AT77C104B
12	TPP	P	Tensión de Estabilización de Temperatura
13	MISO	O	Línea Transmisión Datos del AT77C104B
14	SCANEN	I	Reservado para Test de fabricación, NC
15	SSS	I	Línea Habilitación SSPI, Activo Bajo
16	IRQ	O	Línea de Interrupción hacia el Host, Activo Bajo
17	FSS	I	Línea Habilitación FSPI, Activo Bajo
18	RST	I	Control del RESET y Modo SLEEP, Activo Alto
19	FPL	I	Plano Frontal, conectado a GNDD

Tabla 3.1: Descripción pines del sensor AT77C104B.

El interfaz de control del AT77C104B, recibe comandos a través del Slow SPI para programar, controlar y configurar el sensor.

Siempre que se quiera hacer uso del interfaz de control debe hacerse a una velocidad máxima de 200 KHz y activar el /SSS.

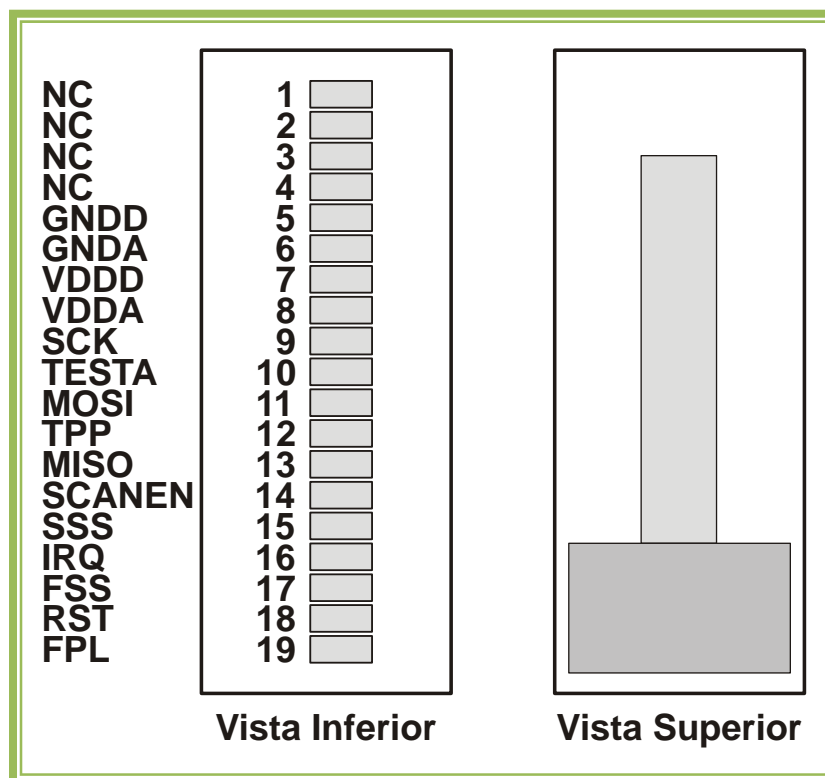


Figura 3.3: Pineado del AT77C104B.

El sensor AT77C104B dispone de un grupo de registros que se utilizan para configurar distintos parámetros y habilitar modos de funcionamiento.

El registro *STATUS* se utiliza para conocer el estado del sensor, y debe ser leído cada vez que este genere una interrupción a través de la línea */IRQ* con objeto de identificar la causa de la interrupción. La Tabla 3.2 muestra un listado de todos estos registros.

Cada uno de los registros internos del sensor esta unívocamente identificado por una dirección de 4 bits. Cuando el host quiere leer o escribir uno de los registros del sensor, lo hace a través de un comando que indica el tipo de operación que desea realizar, lectura o escritura, y la dirección del registro sobre el que quiere actuar.

Registro	Dirección(b3 a b0)	Tipo de Acceso	Descripción
STATUS	0000	Lectura	Muestra el estado del sensor, debe ser leído cada vez que el sensor genera una interrupción para identificar la causa y borrar el flag.
MODECTRL	0001	Lectura/Escritura	Para Programar el modo de trabajo del sensor.
ENCTRL	0010	Lectura/Escritura	Para habilitar las posibles fuentes de interrupción del sensor.
HEATCTRL	0011	Lectura/Escritura	Para habilitar la función de aumento de temperatura del sensor y su potencia.
NAVCTRL	0100	Lectura/Escritura	Para configurar parámetros relativos al desplazamiento del dedo por el sensor.
CLICKCTRL	0101	Lectura/Escritura	Para configurar parámetros relativos a la detección del dedo sobre el sensor.
MOVCTRL	0110	Lectura/Escritura	Para configurar el umbral de detección de movimiento sobre el sensor.
NAVIGATION	1000	Lectura	Utiliza un protocolo similar al PS/2, los tres registros se usan para detectar movimientos sobre los ejes x e y, así como detección de click. No se permite la lectura individual de estos registros, sino que la lectura del primero devuelve el valor de los tres.
NAVIGATION	1001	Reservado	
NAVIGATION	1010	Reservado	

Tabla 3.2: Listado de los registros del AT77C104B.

Estos comandos de operativa y acceso a los registros del sensor consisten en la transferencia de 1 byte a través del Slow SPI. La Figura 3.4 muestra el formato de estos comandos.

La transferencia siempre envía el bit más significativo (MSB) primero. Cuando el MSB (b7) tiene valor '1' indica que este byte corresponde a un comando, mientras que cuando se envía un '0' se asocia al envío de un dato. El b6 en el caso de envío de

comandos, indica si este es de lectura '1' o escritura '0'. Los siguientes cuatro bits (b5-b2) contienen la dirección del registro sobre el que se desea operar, mientras que b1 y b0 carecen de utilidad.

b7	b6	b5	b4	b3	b2	b1	b0
1	Lectura (1) Escritura (0)	Dirección (b3)	Dirección (b2)	Dirección (b1)	Dirección (b0)	X	X

Figura 3.4: Formato de comando.

El número de bits utilizados para escribir los datos en el registro es 7, debido a que el MSB (b7) del byte enviado esta a '0' indicando que es un comando de datos. El formato del byte de datos se muestra en la Figura 3.5.

b7	b6	b5	b4	b3	b2	b1	b0
0	Datos (b6)	Datos (b5)	Datos (b4)	Datos (b3)	Datos (b2)	Datos (b1)	Datos (b0)

Figura 3.5: Formato de datos.

El AT77C104B permite implementar hasta seis modos de operación según las necesidades a cubrir y configurables a través de los registros de control y estado, estos modos son:

- **Sleep mode:** Modo de muy bajo consumo controlado por el pin de reset (*RST*). En este modo, todos los relojes internos están desactivados y los registros inicializados.
- **Stand-by mode:** Modo de bajo consumo que espera una acción desde el host. El interfaz Slow SPI y los bloques de control están activos, así como el oscilador.
- **Click mode:** Modo de espera que detecta cuando se ha puesto un dedo sobre el sensor, permite despertar el dispositivo para nueva lectura o para simular el click del ratón. El Slow SPI, los bloques de control, el oscilador, la matriz de click y el bloque click están activos.
- **Navigation mode:** Modo que detecta el movimiento horizontal y vertical de dedo sobre el sensor, se emplea cuando se quiere usar el sensor a modo de ratón para controlar el movimiento del cursor. El Slow SPI, los bloques de control, el oscilador, la matriz de navegación y el bloque de navegación están activos.
- **Acquisition mode:** Modo en el que los frames de huella dactilar son enviados al host para su reconstrucción e identificación. El interfaz SPI Slow y los bloques de control están activos, así como el interfaz Fast SPI, la matriz de adquisición y el oscilador.
- **Test mode:** modo reservado para configuración en fabricación.

Los modos utilizados en el proceso de adquisición de huellas dactilares se pueden resumir en:

- **Stand-by mode:** Bajo consumo.
- **Pointing mode:** Equivale a los modos Click y Navigation.
- **Acquisition mode:** Para capturar la imagen de huella dactilar.

3.5 Ventajas del Sensor

Tal y como se ha ido presentando a lo largo de este capítulo, las ventajas de esta tecnología de sensor son la consecuencia de su técnica de sensores térmicos, junto con el método de captura de imagen por desplazamiento y la integración del sensor junto con la circuitería de reconstrucción de imagen en un único circuito integrado o IC.

El enfoque térmico no requiere ninguna actuación sobre el dedo, solo hace uso de las propiedades fisiológicas de una yema viva, reduciendo así el consumo, y cualquier molestia al usuario derivada de acondicionar el dedo con corriente o radiofrecuencia.

El método de desplazamiento reduce el área necesaria para la captura de la imagen en un factor de 5, reduciendo de forma proporcional su coste. La imagen reconstruida es de alta resolución. Este método, también implica que la ventana del sensor es autolimpiable, impidiendo la posibilidad de que queden huellas latentes tras una captura. Además se ha podido probar que resulta verdaderamente difícil desplazar un dedo artificial con la suficiente suavidad, como para que la imagen capturada pueda ser reconstruida.

El haber integrado en el mismo IC la circuitería necesaria para la reconstrucción de la huella, no solo ha reducido costes del desarrollo final de todo el sistema y consumo, sino que además incrementa la velocidad de trabajo del sistema biométrico.

Capítulo 4

Plataforma de Desarrollo

La plataforma de desarrollo final de este proyecto, es el resultado de las distintas plataformas hardware utilizadas. La selección de esta combinación hardware ha sido condicionada por los resultados obtenidos desde la plataforma inicial, la accesibilidad de los sensores de huella dactilar en el mercado, así como la intención de aprovechar los recursos ya disponibles. Esta evolución hasta el desarrollo final, se puede resumir en las siguientes tres combinaciones hardware:

- Sensor de Huella Dactilar Fujitsu MFB310v5 con Kit de desarrollo Keil MCB2100.
- Sensor de Huella Dactilar Atmel AT77C104B con Kit de desarrollo Keil MCB2100.
- Sensor de Huella Dactilar Atmel AT77C104B con Kit de desarrollo Embest LPCEB2000-B sobre LPCEB2000-I.

A continuación se justifica cada una de ellas y se resumen sus principales características e inconvenientes.

4.1 MFB310v5 con MCB2100

Esta combinación era la propuesta inicial del proyecto, se disponían tanto del kit de desarrollo MCB2100, mostrada en la Figura 4.1 como de una placa de circuito integrado, diseñada para el sensor de huella dactilar MFB310v5 de Fujitsu.

La placa de evaluación incluye un microcontrolador de arquitectura ARM7TDMI-S de 32 bits, el LPC2129 de NXP; un cristal de 12 MHz, que le permite trabajar hasta a 60 MHz, cuatro conectores DB9, 2 para el CAN y 2 para cada uno de los dos interfaz RS232, un interfaz JTAG, botón de reset e interrupción, leds y acceso a todos los puertos de propósito general.[12]

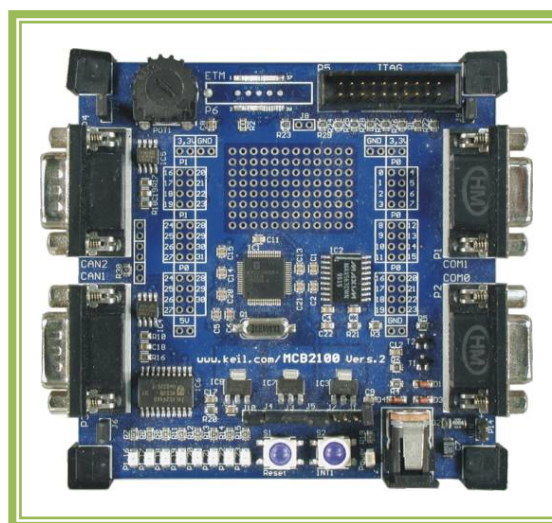


Figura 4.1: Vista superior MCB2100.

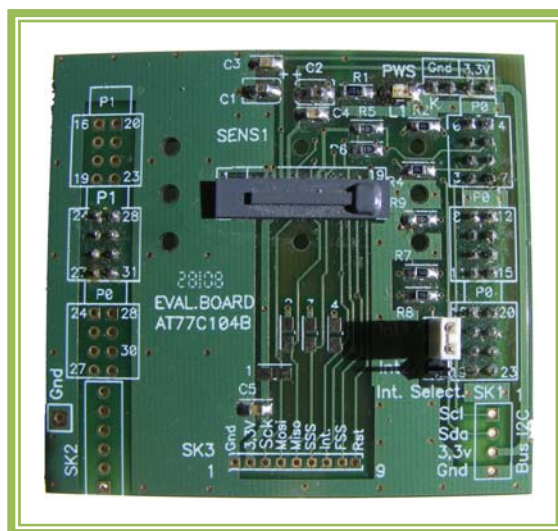
El sensor de huella dactilar por desplazamiento de Fujitsu,[13] MBF310v5 es un dispositivo de adquisición de huella dactilar por desplazamiento capacitivo. Cada frame es representado por 218x8 pixeles con una resolución de 500 dpi, dispone de dos interfaces de comunicación con el host, uno implementando un bus de 8 bits y el otro a través de un interfaz SPI, permitiendo velocidades de transmisión de datos de hasta 20 Mbps.

El desarrollo parecía viable a priori, ya que se disponía de los dispositivos e interfaces apropiados para que existiera comunicación entre ellos, optando por el interfaz SPI, para control y adquisición de datos y una UART para el envío de los datos recogidos relativos a la imagen, para su posterior procesamiento y reconstrucción en un PC. Sin embargo, el proceso de desarrollo se vio frenado debido a la imposibilidad de comunicación con el sensor debido a la escasa información disponible del sensor.

4.2 AT77C104B con MCB2100

Consecuencia de la mala experiencia con el dispositivo de Fujitsu, aparece el AT77C104B, resultado de la búsqueda de un sensor de huella dactilar por desplazamiento, con características similares al MBF310v5 en cuanto a interfaces de comunicación con el objetivo de poder aprovechar los conocimientos adquiridos en el manejo de kit de desarrollo MCB2100.

En el capítulo anterior se ha presentado el AT77C104B, un sensor de huella dactilar térmico por desplazamiento, que será el usado para la implementación del sistema de captura y reconstrucción de huella dactilar con sensor de desplazamiento, para lo cual se ha diseñado una placa de circuito integrado, EVAL.BOARD AT77C104B, mostrada en la Figura 4.2, compatible con la expansión de los puertos de propósito general del kit de desarrollo MCB2100. Los puertos e interfaces del sensor necesarios para poder comunicar, configurar y adquirir datos, son:



**Figura 4.2: Vista superior
EVAL.BOARD AT77C104B.**

- Una Línea de Reset, *RST*, que permita inicializar el sensor, línea de entrada al sensor.
- Interfaz SPI, con un único interfaz SPI desde el host, se pueden satisfacer la necesidades de configuración y adquisición del sensor.

- Líneas de control Slow SPI y Fast SPI, necesarias para comunicar al dispositivo si esta en modo configuración; /SSS; o Adquisición; /FSS. Estas líneas son de entrada al sensor.
- Una línea *IRQ*, que permita al sensor indicar al host algún evento inesperado, en nuestro caso, detección del dedo y su movimiento, esta línea es de salida desde el sensor.

En principio, esta plataforma es perfectamente compatible, pero no dispone de suficiente memoria para almacenar una captura. Sin embargo, la solución que se describe en el apartado siguiente permite además de reutilizar el código desarrollado para esta solución, aprovechar las características del AT77C104B.

4.3 AT77C104B con LPCBE2000-B

Esta alternativa es un kit de desarrollo de Embest, el LPCEB2000-I, es una tarjeta de expansión que proporciona entre otros, 3 conectores DB9, 2 para puertos serie y 1 para puerto CAN, un interfaz JTAG, botón Reset, teclado 4x4, 8 leds, 8 displays de 8 segmentos a los módulos insertados en el puerto de expansión de la placa LPCEB2000-I.

De los tres módulos de Embest compatibles con la tarjeta expansora LPCEB2000-I, se dispone de el modulo LPCEB2000-B, Figura 4.3, provisto de un microcontrolador de arquitectura ARM7TDMI-S de 32 bits, el LPC2292 de NXP, un cristal de 10 MHz, permitiendo velocidades de hasta 50 MHz para la CPU, 2 UART, 2 Interfaces SPI, 1 Interfaz CAN, acceso a todos los puertos de propósito general, memoria Flash externa de 1M x 16 bit modelo AT49BV162A y memoria SRAM externa de 256K x 16 bit modelo IS61LV25616.[14]

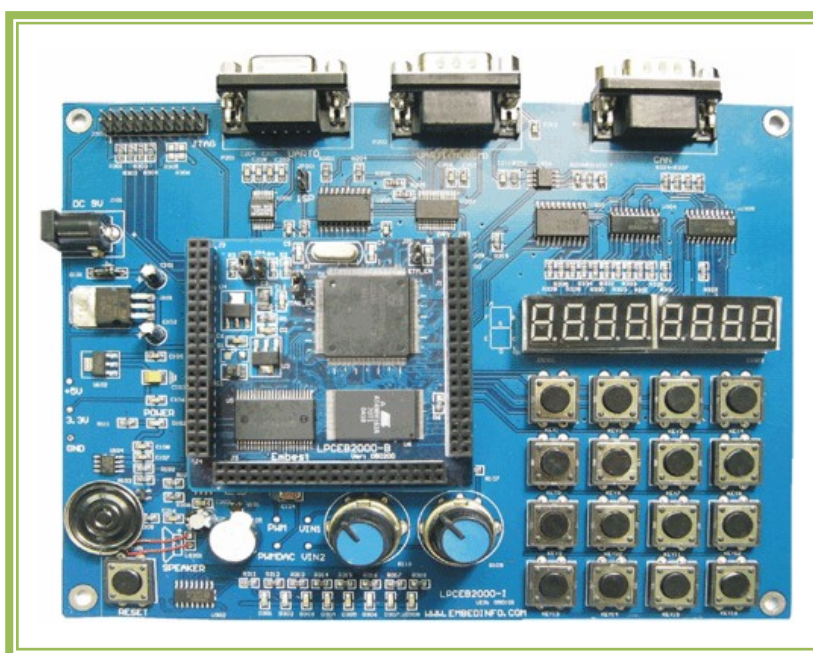
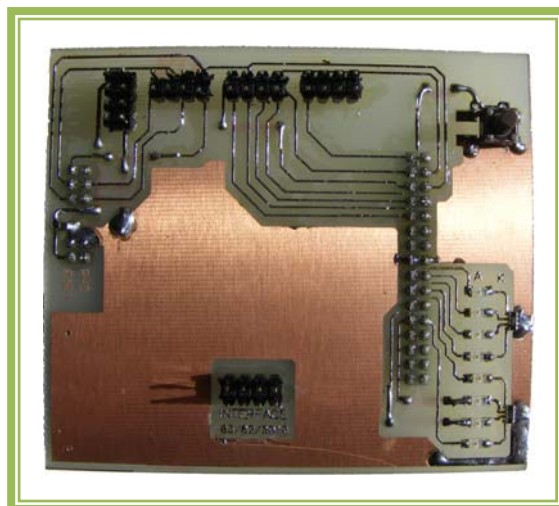


Figura 4.3: Vista superior de LPCEB2000-B sobre LPCEB2000-I.

Este sistema microcontrolador es compatible con el descrito en el apartado 4.1. Aportando como novedad la expansión de memoria, y por tanto el software desarrollado para la combinación descrita en el apartado anterior es válido.

Los puertos e interfaces necesarios para controlar el sensor de huella dactilar y la recepción de datos del sensor, así como el envío al PC de la imagen para la representación en pantalla, identificación o autenticación, son:

- Línea de Interrupción para atender las peticiones del sensor, Se configura el pin de propósito general P0.14 para *EINT1*.
- Línea de Reset, *RST*, para inicializar el sensor a través del pin P0.12 y configurada como salida.
- Líneas de control para el */SSS* y */FSS*, se utilizan los pines de propósito general P0.10 y P0.11, configurados como salidas.
- Un interfaz SPI, concretamente el *SPI0*, para la toda la comunicación serie entre el micro y el sensor, tanto en modo */SSS* y */FSS*. La transmisión SPI, *MOSI0*, se realiza por el pin P0.6; la línea de recepción del interfaz SPI, *MISO0*, se lleva a acabo en el pin P0.5; la señal de reloj que controla la comunicación SPI, *SCK0*, se inserta por el pin P0.4; finalmente la línea que habilita el interfaz, *SSEL0* en el pin P0.7 es controlada a través del pin P0.13 configurado como salida.
- Se configuran como salidas 8 puertos, P1.16 - P1.23, para activar leds, que se utilizaran para información visual al usuario.
- El modulo de memoria expandida SRAM, para almacenar la huella dactilar recibida.
- Un interfaz RS232, para la comunicación serie con el PC, para en envío de la huella dactilar a representar, concretamente la *UART0*, a través de los pines de propósito general P0.0 y P0.1 para *TXD0* y *RXD0* respectivamente.



**Figura 4.4: Vista Superior
INTERFACE.**

Debido a que la distribución de los pines de propósito general en LPCBE200-B es distinta a la de la MCB2100, y que la EVAL.BOARD AT77C104B ha sido diseñada para la disposición de GPIO de esta ultima, se ha realizado la placa interfaz, INTERFACE, Figura 4.4, para hacer compatible la placa de circuito integrado del AT77C104B con la LPCBE2000-B, dotándola además de 8 leds conectados a través de dos arrays de resistencias de 220 Ω a los pines P1.16 - P1.23 y un pulsador conectado a P0.15. Resultando el esquema de interconexión de la plataforma final el mostrado en la Figura 4.5.

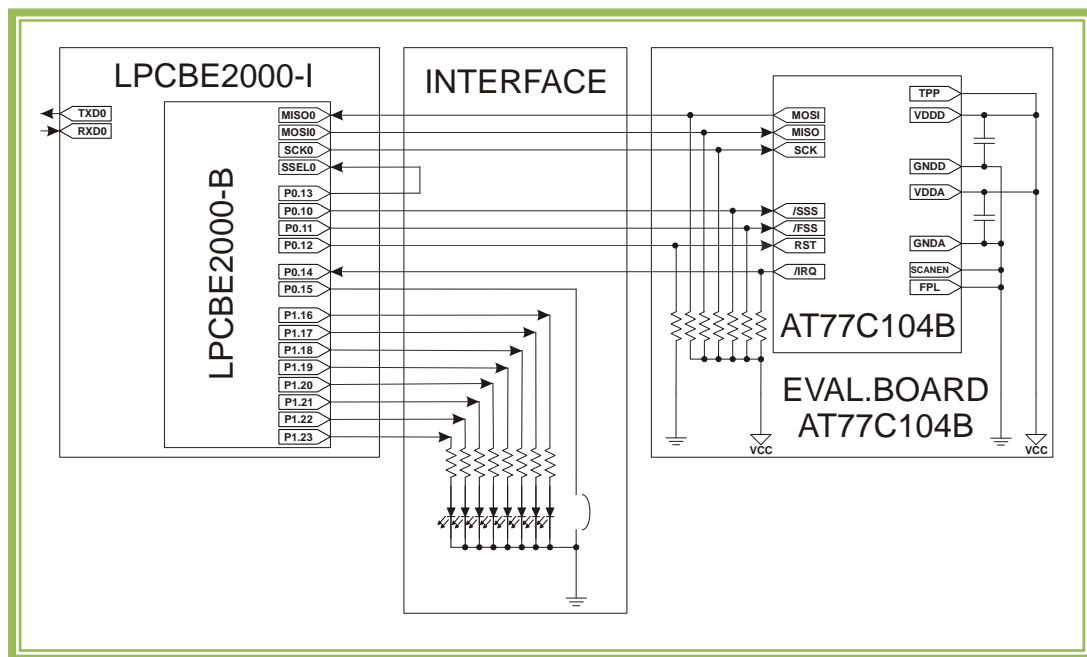


Figura 4.5: Diagrama de bloques de sistema desarrollado.

Como se ha explicado en el capítulo anterior, el modo de adquisición de imagen del sensor de huella soporta velocidades de transferencias máximas de hasta 16 Mbps, cuanto mayor velocidad en la transferencia de datos hacia el host, mayor velocidad de desplazamiento es soportada por el sensor. Como se ha mencionado anteriormente la velocidad de trabajo de la CPU es de 50 MHz, como este reloj controla la velocidad del reloj del interfaz SPI según la relación $SPICLK = PCLK / SPCCR$, siendo $PCLK = CCLK / VPB$ divider y $SPCCR \geq 8$. Parece conveniente trabajar a la mayor velocidad posible para reducir en lo posible las limitaciones del sistema, con esta premisa la máxima velocidad de trabajo del interfaz SPI es de 6,25 MHz, resultado de un CCLK o reloj del sistema de 50 MHz, con un VPB divider = 1, el reloj de los periféricos, PCLK será de 50 MHz y como el mínimo valor del SPCCR que se admite es 8,[15] el cociente entre 50 MHz y 8 resulta en 6,25 MHz como máxima velocidad del interfaz SPI en modo adquisición o Fast SPI.

Fsck (MHz)	Tasa Transferencia (Mbps)	Tasa de Frames (Frames/Sg)	Velocidad Max. Desplazamiento (Cm/Sg)	Relación
1	1	134	3	Muy Lenta
2	2	268	6	Muy Lenta
4	4	536	12	Mínimo
6	6	804	18	Normal
8	8	1072	24	Buena
12	12	1608	36	Muy Buena
16	16	2146	48	Muy Buena

Tabla 4.1: Velocidad de desplazamiento frente frecuencia de transmisión.

En la Tabla 4.1,[11] se muestra la relación existente entre la velocidad de desplazamiento del dedo sobre la superficie del sensor y el reloj de transferencia de datos, relación que es necesario cumplir para una buena reconstrucción y limitando el sistema a

velocidades de desplazamiento del dedo máximas de en torno 18 Cm/Sg que atiende a una velocidad normal.

Capítulo 5

Descripción de la Solución

El objeto de este capítulo, es explicar como se ha desarrollado la solución para el control del sensor, adquirir los datos relativos a la huella dactilar y la ordenación de los mismos para la posterior representación de la huella.

En primer lugar se tratará una serie de consideraciones generales sobre el diseño software, su comunicación para representación en pantalla y las medidas adoptadas.

Posteriormente se presenta una escueta explicación del código desarrollado.

5.1 Consideraciones Generales

Asumiendo que la velocidad de desplazamiento del dedo sobre la superficie del sensor se hace a suficiente velocidad dentro de los márgenes indicados por el fabricante y siendo adecuada la velocidad de trabajo máxima del interfaz SPI, aprovechando la circuitería del AT77C104B se puede obtener una imagen clara y completa de una huella. Lo siguiente a tener en cuenta parece ser la capacidad de detección de inicio de un dedo desplazándose sobre la superficie del sensor y cuando ha terminado de hacerlo. El sensor dispone de los registros necesarios para poder discriminar estas situaciones, pero la capacidad de memoria de almacenamiento de la que dispone la LPCEB2000-B, desaconseja este método, para evitar un posible desbordamiento de memoria.

Una huella dactilar promedia una longitud de 30 mm, sabiendo que el tamaño de la ventana de captura del AT77C104B es de 0,4 mm, son necesarias 75 muestras consecutivas para tener una imagen completa. Desafortunadamente, las muestras se toman consecutivas según se desliza el dedo por la superficie del sensor, lo que resulta en que frames consecutivos repitan información. Planteando la posibilidad de que el solapamiento afecte al 50% de la muestra, parece lógico que recogiendo una lectura de 150 - 160 muestras se tiene suficiente información para obtener imágenes completas.

Partiendo de esta premisa, calculando el espacio en memoria de 160 muestras, tenemos:

- Cada frame ocupa 8 filas x 232 columnas, lo que hace 1856 pixeles/frame, a los que hay que sumarle 8 pixeles/frame de cabecera acumulando un total de 1864 pixeles/frame. Según se muestra en la Figura 5.1.[11]
- Cada pixel es representado por 4 bits, por tanto cada frame ocupa 7456 bits, multiplicados por los 160 frames necesarios para reconstruir una imagen completa hacen 1192960 bits.
- La SRAM de la que dispone la placa LPCBE200-B es de 256Kx16bit, el total de bits que puede almacenarse en esta memoria es de $2^{18} \times 16 \text{ bits} = 4194304 \text{ bits}$, lo que induce pensar que se podrían almacenar hasta tres imágenes completas en la memoria del sistema.

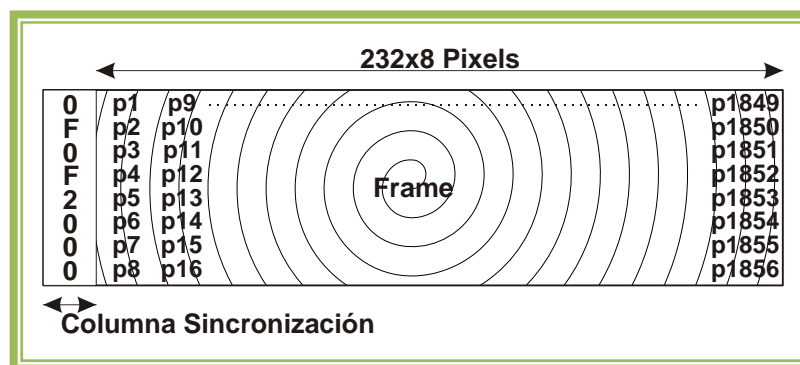


Figura 5.1: Disposición de pixels en un frame.

Desafortunadamente aunque los datos almacenados corresponden a una imagen completa, estos presentan aun dos inconvenientes. El primero es que no están ordenados y disponen de la cabecera de cada frame, según se muestra en la Figura 5.2,[11] esto obliga a reservar espacio de memoria para los datos ordenados y libres de la cabecera.

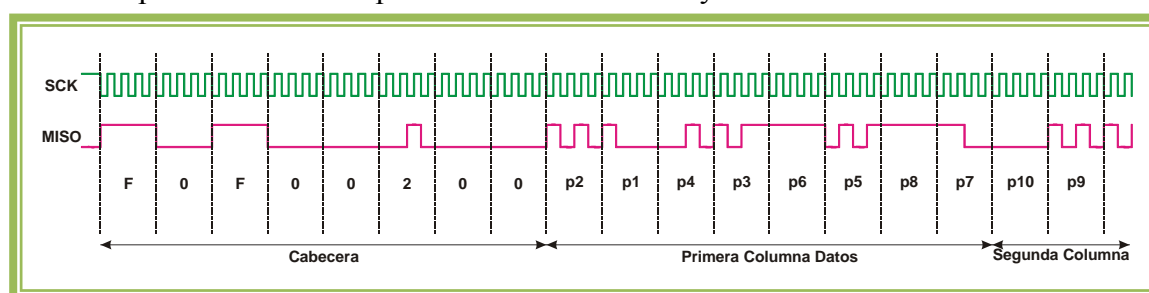


Figura 5.2: Cronograma recepción de imagen.

El segundo inconveniente es que cada pixel esta representado con 4 bits y la representación habitual en escala de grises es de 8 bpp, por lo que se plantea necesario además cambiar la profundidad de pixel de 4 bits a 8 bits, lo que implica doblar el tamaño de la imagen completa, pasando de 1187840 bits (1192960 bits menos 5120 bits de cabeceras) a los a 2375680 bits que junto con los 1192960 bits que ya están almacenados suman 3568640 bits lo que limita la memoria a una única captura, por tanto no se permite una nueva lectura sin haber enviado previamente la ultima adquirida. Este cambio de profundidad de pixel supone el otro inconveniente, resulta en una imagen poco contrastada pues solo se tienen los 16 primeros tonos de grises consecutivos desde el negro hacia el blanco, así que para darle mayor contraste a la imagen habría que aplicar algún algoritmo a cada pixel para hacer un ajuste de nivel y aprovechar el rango del espectro de grises.

Una vez se tiene almacenada la huella dactilar se debe enviar a un PC, para procesarla como un archivo de imagen y presentarla en pantalla. Para ello se ha desarrollado una aplicación en C#, que permanece a la escucha del puerto serie configurado a la espera de recibir datos, una vez se reciben los datos, se identifican por medio de una cabecera al comienzo de la imagen si es un texto informativo para representación en pantalla o la imagen para su procesamiento. Cuando se detecta la cabecera de inicio de imagen consistente en el envío de 2 bytes y representada por los caracteres ASCII 'yz' los datos recibidos se almacenan en un archivo hexadecimal con extensión RAW, hasta que se reciben los 2 bytes de fin de imagen representados por los caracteres 'rs'.

Para poder obtener un archivo de imagen representable, se ha utilizado Infarview, un visor grafico freeware, que permite procesar archivos binarios en imagen indicándole la profundidad de bit y las dimensiones en pixeles de la imagen. Estos parámetros se ejecutan en segundo plano, así como el almacenamiento de la imagen resultante en formato JPEG, de forma que resulta transparente para el usuario.

Una vez se tiene el archivo JPEG, se muestra por pantalla, tal y como se muestra en la Figura 5.3.

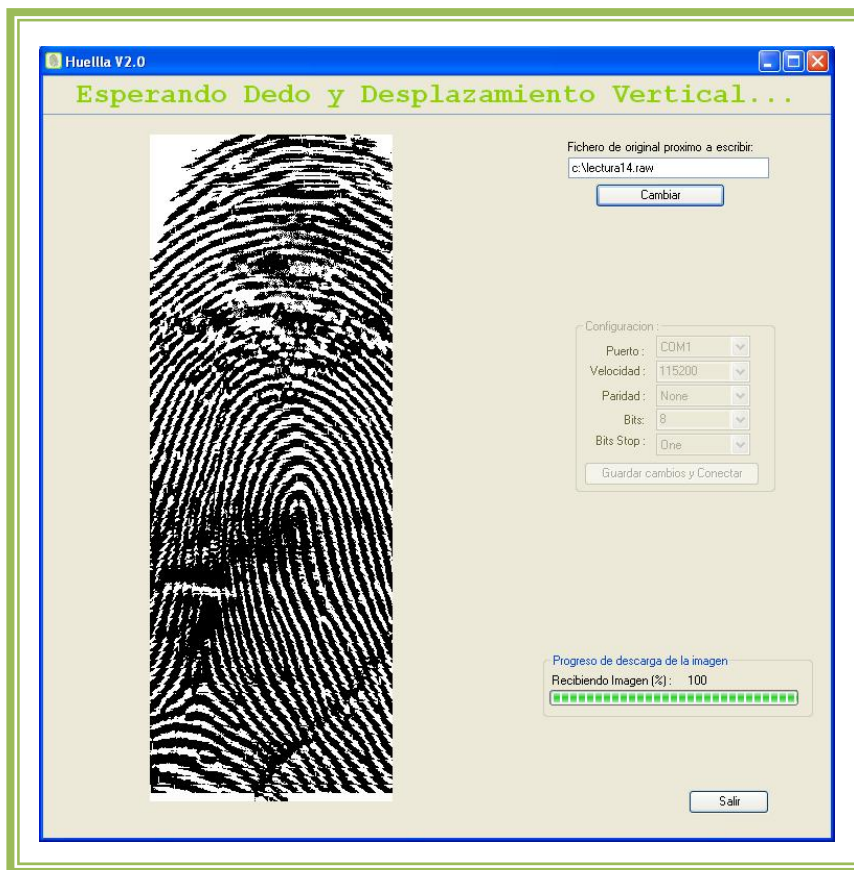


Figura 5.3: Imagen de huella capturada.

5.2 Solución Software

Tal y como se ha mencionado anteriormente, la plataforma de desarrollo sobre la que se ha diseñado la solución software es la formada por el sensor AT77C104B y el kit del LPCEB2000-B.

Tras estudiar las principales características del sensor y el microcontrolador, compatibilidad, ventajas e inconvenientes, se afronta el diseño software de acuerdo a las consideraciones mencionadas en capítulos anteriores. A continuación, se presenta brevemente el código de la aplicación y se describen las decisiones más relevantes de este diseño, en el apéndice A se muestra la solución completa:

- Habilitar IRQs.

Configura y habilita la Interrupción Externa *EINT1*, activa por flanco de subida a través del pin P0.14 para solicitar al microcontrolador su atención a una petición originada en el sensor.

- Inicializar PLL.

Atendiendo a las especificaciones del fabricante se configura el PLL a través del *PLLCFG* con un valor del Multiplicador de 5 ($MSEL = 5$) y del Divisor de 2 ($PSEL = 2$), que partiendo de un oscilador externo de 10 MHz proporciona una frecuencia de trabajo para la CPU de 50 MHz. se activa el PLL escribiendo en el registro *PLLCON* el valor correspondiente, y tras proporcionar la secuencia de inicialización del PLL se habilita de nuevo escribiendo en *PLLCON*.

Una vez el PLL esta activo, se escribe el registro *VPBDIV* para establecer el reloj de los periféricos a la misma frecuencia que el reloj del sistema, de esta manera se podrá disponer de la máxima frecuencia de trabajo para el bus SPI.

- Inicializar SPI y GPIO.

Se proporcionan las instrucciones necesarias para disponer del *SPI0*, la *UART0*, la interrupción externa *EINT1* y configurar el sentido de los pines de propósito general necesarios para controlar el sistema y se programa su estado.

El LPC2292 actúa como maestro en la comunicación SPI, para ello la línea *SSEL0* se establece a nivel alto a través del P0.13 para disponer en todo momento del reloj para la comunicación, que se fija en 200 KHz, además es necesario configurar la fase y polarización de este reloj ($CPOL = 0$, $CPHA = 1$) para que exista entendimiento entre el microprocesador y el AT77C104B. La transferencia SPI se controla por pulling y se envían los bits más significativos primero.

- Inicializar UART.

Configura y habilita la *UART0* para trabajar a 115200 Bps, 8 bits, sin paridad y 1 bit de stop.

- Inicializar SRAM.

Habilita la SRAM externa según la rutina de inicialización de memoria, y se inicia puntero al inicio de la parte de memoria de los datos adquiridos.

- Inicializar Sensor.

Antes de realizar cualquier operación sobre el AT77C104B, este debe ser inicializado, esta inicialización consiste en la activación de la línea *RST* del sensor por un tiempo superior a 10 uSg. Se debe tener la precaución de mantener la líneas */SSS* y */FSS* a nivel alto durante la secuencia de inicio del sensor para evitar que este entre en modo *SCAN TEST*.

Una vez finalizada la secuencia de inicialización del sensor, es necesario un tiempo de 1 uSg (T_{rstsu}) antes de que el puerto SPI del sensor este disponible.

- Configurar el sensor a través de los registros necesarios usando Slow SPI.

Se establece el modo de trabajo del sensor a través del interfaz de control sensor.

Los códigos de operación, direcciones de los registros del sensor y los datos para ser escritos o leídos se pasan a través de las funciones *escribeRegistro(codigoOperacion, dirección)* y *leeRegistro(codigoOperacion)*, siendo *codigoOperacion* y *dirección* variables globales. Una vez se ha habilitado el */SSS* y habiendo esperado un T_{sssu} mínimo de $\frac{1}{2} T_{sck}$ (1 uSg) se envía el primer byte (operación y dirección) al sensor a través del SPI realizando seguidamente una lectura del buffer de recepción para liberar el flag y continuar con el envío del segundo byte con los parámetros de configuración

del registro en cuestión. Una vez se ha completado la secuencia de escritura del sensor, se deshabilita el /SSS respetando una vez mas el T_{sshd} .

- Pasar el sensor a *modo Navigation*, habilitando su interrupción.

Una vez se ha configurado el sensor térmico con los parámetros de trabajo deseados, se prepara el sensor para generar una interrupción cuando detecte movimiento en su superficie.

- Pasar a *modo Adquisición*.

Cuando el host determina, tras la lectura del registro *STATUS*, que la interrupción ha sido causada por la interrupción registro *NAVIGATION*, se configura el sensor en *modo Adquisición*.

El *modo Adquisición* se habilita haciendo un ciclo de escritura sobre el registro *MODECTRL*.

Se configura el sensor para generar una interrupción en caso de error de lectura para ser informado el host en caso de producirse un error durante la lectura de datos y poder responder debidamente ante esta situación

- Configurar Fast SPI.

Para poder recibir los datos desde el sensor a una tasa adecuada, es necesario configurar de nuevo la velocidad de trabajo del *SPI0*, para lo que se emplea la función *configuraFSS()* que escribe los registros necesarios para configurar la velocidad de trabajo del bus a 6,25MHz, previamente se deshabilita /SSS y /FSS.

- Adquirir Datos.

La adquisición de datos comienza con la llamada a la función *adquisicionDatos()*, que se muestra en la Figura 5.4 y se explica a continuación.

Se inicializa el puntero para el acceso a memoria SRAM a la dirección en la que comienza la recepción de los datos enviados por el sensor.

Se habilita el modo Fast SPI poniendo a cero la línea /FSS. Desde este momento el sensor comienza a enviar datos hacia el host, se debe tener en cuenta que antes de que los primeros datos se reciban, concretamente el primer byte de la cabecera, se deben de procesar 40 ciclos de reloj.

Los 160 frames que se estima conveniente adquirir, hacen un total de 298240 pixeles (160 frames x 233 columnas x 8 filas) como en cada transferencia SPI se envían hacia el host 2 pixeles, y cada posición de memoria puede contener 4 pixeles, solo hay que escribir en memoria cada dos datos recibidos desde el sensor.

Una vez finalizada la adquisición de datos, se debe deshabilitar el /FSS.

- Confirmar que no existen errores.

Si han ocurrido errores de lectura durante la transferencia de información desde el sensor térmico hacia el host, se notifica y solicita una nueva lectura.

- Ordenar Datos.

En caso de que la lectura de datos este libre de errores, se procede a la ordenación de los datos recibidos mediante la función *ordena()* que se muestra en la Figura 5.5.

Los datos ordenados, se almacenan en la memoria SRAM, a partir de la dirección 0x81012360.

```

void adquisicionDatos()
{
    unsigned long i = 0;
    unsigned char Data = 0x00;           // Para Recepción de Datos
    unsigned short temp = 0x0000;        // Para Datos Memoria
    ledOr = 0x00010000;
    led = ledOr;
    pSRAM = (unsigned short *)INICIO_SRAM; // Inicializa Puntero Memoria Ordenado
    IO0CLR = FSS;                         // Habilita Fast SPI

    do                                   // Espera los 40 ciclos de reloj hasta
    {                                   // primer dato
        SOSPDR = DUMMY;
        finTx();
        Data = SOSPDR;
    } while((Data != 0xF0)); //endDw    // Data contiene el primer 0xF0

    for(i=0; i < 149121; i++)           // 298240 pixels para 160 frames (160 x
                                        // 233(232+Dummy) x 8 columnas), 2 pixels por
                                        // vuelta
    {
        if(i%2)                        // Si es lectura impar
        {
            temp = temp | Data;         // Byte bajo de la dirección de memoria (2
                                        // pixels)
            escribeSRAM(pSRAM, temp);   // Se almacenan los últimos 4 pixels
                                        // recibidos
            pSRAM++;                    // Incrementa 1 Dirección de Memoria
        } //endif
        else                            // Si es lectura par
        {
            temp = Data;
            temp = temp << 8;           // Byte alto de la dirección de memoria (2
                                        // pixels)
        } //ende
        SOSPDR = DUMMY;
        finTx();
        Data = SOSPDR;
        if(!(i % 18640) | (i == 149120)) // Leo siguientes 2 pixels
                                        // Para Información visual
        {
            IO1SET = led;
            led = led << 1;
            led = led | ledOr;
        } //endif
    } //endf
    IO0SET = FSS;                       // Deshabilita Fast SPI
    espera(uSgl);                        // Para respetar Tsshd
} //end adquisicionDatos

```

Figura 5.4: Función adquisicionDatos().

Como la memoria esta orientada a 16 bits, cada dirección de memoria contiene los cuatro primeros pixeles o los cuatro últimos pixeles de cada columna del frame, incluido los de cabecera; además están organizados de forma tal que esta alterado el orden de los pixeles pares con los impares, tal y como se muestra en la Tabla 5.1.

DIRECCIÓN MEMORIA	B15-B16	B11-B8	B7-B4	B3-B0	COMENTARIOS
0x81012360	0xF	0x0	0xF	0x0	B0-B4 CABECERA
0x81012361	0x0	0x2	0x0	0x0	B5-B7 CABECERA
0x81012362	P02	P01	P04	P03	B0-B4 1ª COLUMNA
0x81012363	P06	P05	P08	P07	B5-B7 1ª COLUMNA

Tabla 5.1: Contenido 4 primeras direcciones de memoria recepción.

```

void ordena()
{
    unsigned short temp, temp1, temp2, temp3;

    unsigned char i, j, k;
    pOrdenados = (unsigned short *)ORDENAR_SRAM;
    pSRAM = (unsigned short *)INICIO_SRAM;
    for(i = 0; i < 160; i++)
    {
        for(j = 2; j < 4; j++)
        {
            pSRAM = (unsigned short *)INICIO_SRAM;
            pSRAM = pSRAM + j + (i*466);

            for(k = 0; k < 116; k++)

            {
                temp1 = 0;
                temp2 = 0;
                temp3 = 0;
                temp = leeSRAM(pSRAM);

                pSRAM = pSRAM + 2;
                temp1 = temp & 0xF00;

                temp = leeSRAM(pSRAM);

                pSRAM = pSRAM + 2;
                temp2 = temp & 0xF00;

                temp2 = temp2 >> 8;

                temp3 = temp1 | temp2;
                temp3 = temp3 * 0x7;
                escribeSRAM(pOrdenados++, temp3);
            }

            pSRAM = (unsigned short *)INICIO_SRAM;
            pSRAM = pSRAM + j + (i*466);

            for(k = 0; k < 116; k++)

            {
                temp1 = 0;
                temp2 = 0;
                temp3 = 0;
                temp = leeSRAM(pSRAM);

                pSRAM = pSRAM + 2;
                temp1 = temp & 0xF00;

                temp1 = temp1 >> 4;

                temp = leeSRAM(pSRAM);

                pSRAM = pSRAM + 2;
                temp2 = temp & 0xF00;

                temp2 = temp2 >> 12;

                temp3 = temp1 | temp2;

                temp3 = temp3 * 0x7;

                escribeSRAM(pOrdenados++, temp3);
            }
        }
    }
}

```

// Para agrupar 2 pixel convertidos a 8 bits en una palabra de 16 bits para almacenar en memoria

*// Puntero a inicio datos adquiridos + j (salvo la columna de inicio de frame) + frame (i*466), cada frame ocupa 466 posiciones de memoria*

*// Primera fila (i*466) si j = 2 o quinta fila (i*466) si j = 3 del frame*

// Contiene P2, P1, P4, P3 si j= 2 o P6, P5, P8, P7 si j = 3

// Incremento a siguiente Columna

// Cojo Pixel P(1+16j) si j = 2 o P(5+16j) si j = 3 de cada frame

// Contiene P10, P9, P12, P11 si j = 2 o P14, P13, P16, P15 si j = 3

// Incremento a siguiente columna

// Cojo Pixel P(9+16j) si j = 2 o P(13+16j) si j = 3 de cada frame

// P(9+16j) si j = 2 o P(13+16j) si j = 3 pasa a ocupar la parte baja de la palabra

// Cambio a 256 grises

// Almaceno ordenado y escalado

*// Puntero a inicio datos adquiridos + j (salvo la columna de inicio de frame) + frame (i*466), cada frame ocupa 466 posiciones de memoria*

*// Segunda fila (i*466) si j = 2 o sexta fila (i*466) si j = 3 del frame*

// Contiene P2, P1, P4, P3 si j = 2 o P6, P5, P8, P7 si j = 3

// Incremento a siguiente columna

// Cojo Pixel P(2+16j) si j = 2 o P(6+16j) si j = 3 de cada frame

// P(2+16j) si j = 2 o P(6+16j) si j = 3 pasa a ocupar la parte alta de la palabra

// Contiene P10, P9, P12, P11 si j = 2 o P14, P13, P16, P15 si j = 3

// Incremento a siguiente columna

// Cojo Pixel P(10+16j) si j = 2 o P(14+16j) si j = 3 de cada frame

// P(10+16j) si j = 2 o P(14+16j) si j = 3 pasa a ocupar la parte baja de la palabra

// Cambio a 256 grises

// Almaceno ordenado y escalado

Figura 5.5: Función ordena().

Lo que se pretende, es aplicar un algoritmo que permita almacenar la imagen, libre de cabeceras, cada pixel representado por 8 bits y ordenados por filas, además de mejorar el nivel de contraste de la imagen, quedando los datos almacenados en memoria según muestra la Tabla 5.2.

DIRECCIÓN MEMORIA	B15-B8	B7-B0
0x81012360	P01	P09
0x81012361	P17	P25
0x81012362	P33	P41
0x81012363	P49	P57

Tabla 5.2: Contenido 4 Primeras Direcciones de Memoria Ordenadas.

- Enviar Datos.

Se recorren las posiciones de memoria donde esta la imagen almacenada y se envía byte por byte, para su procesado y presentación.

- Presentar Huella Adquirida.

La huella puede verse en pantalla usando una aplicación diseñada en C# que una vez se han configurado los parámetros para la comunicación serie, permanece a la escucha del puerto a la espera de datos procedentes del sistema de adquisición de huella, esta aplicación distingue entre los texto relativos a la información del sistema y la propia imagen de huella dactilar, tratando cada uno según corresponda.

- Listo para nueva lectura....

Una vez la imagen se ha presentado por pantalla, el sistema esta listo para una nueva captura.

En la Figura 5.6 se muestra un diagrama de flujo de toda la aplicación.

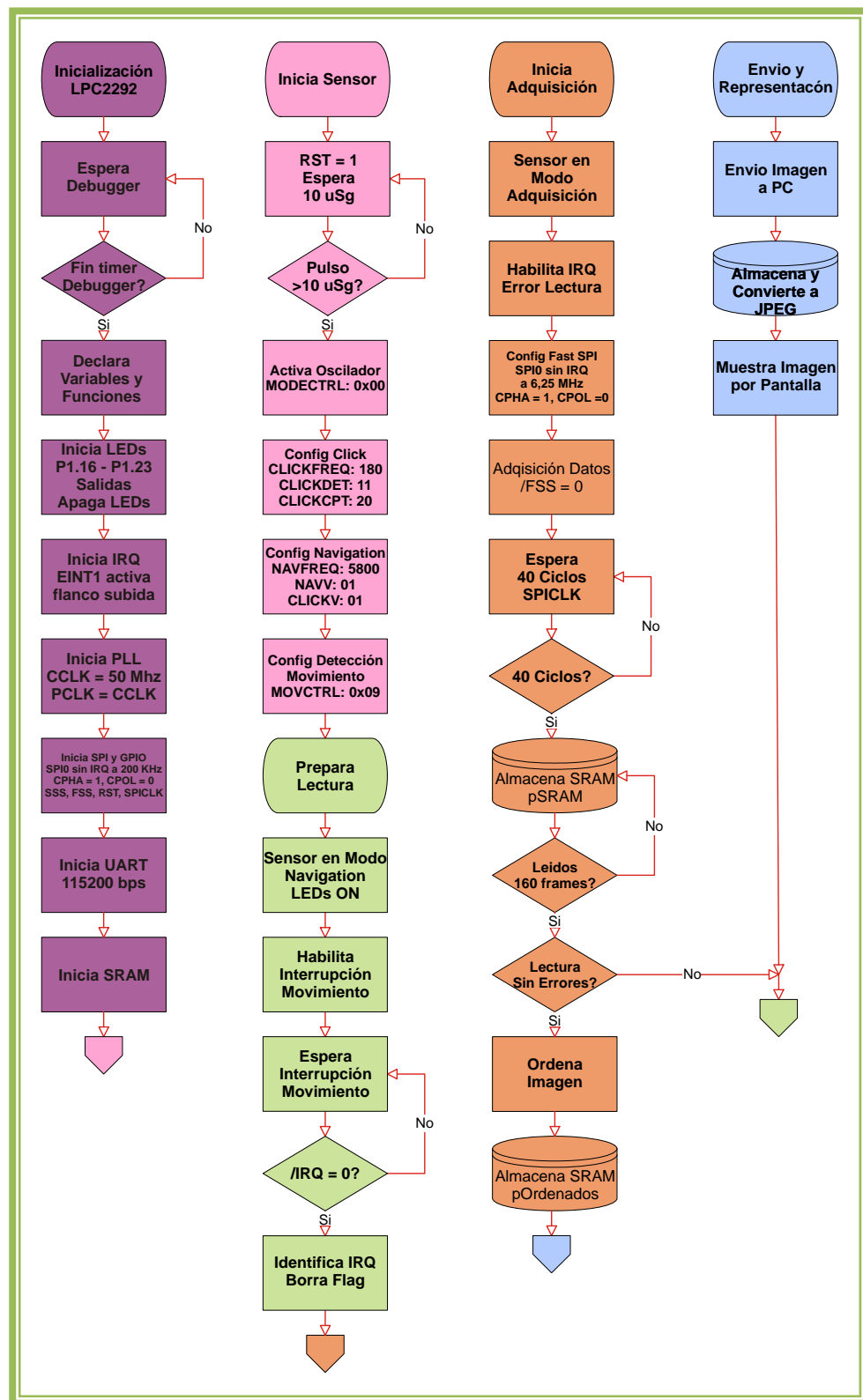


Figura 5.6: Diagrama de flujo de la aplicación.

Capítulo 6

Pruebas

En este capítulo se van a mostrar un conjunto de imágenes capturadas con distintas configuraciones del sensor y distintos parámetros de adquisición y procesado, con el fin de determinar cual es la mejor configuración para nuestro sistema.

6.1 Según el número de frames.

En primer lugar se determina el número de frames necesario para obtener una imagen completa, dando lugar a las consideraciones presentadas en la sección 5.1.

Todas las imágenes que se muestran en la Figura 6.1 representan el total de los datos capturados, por lo que se puede distinguir entre los datos pertenecientes a los ordenados y los descartados por pertenecer a líneas de frame solapadas.

- **Captura de 80 frames.**

Como se ha explicado en la sección 5.1, 80 frames es el mínimo necesario para obtener una imagen completa en cuanto a dimensiones se refiere, siempre que no existiera solapamiento.

Las imágenes obtenidas con este numero de frames, mostradas en la Figura 6.1, imagen *a*), parecen apropiadas para un sistema biométrico, sin embargo la velocidad de desplazamiento del dedo, debe ser lo bastante rápida como para disponer de una porción de huella dactilar suficiente.

Apenas parece que existan líneas solapadas.

- **Captura de 120 frames.**

En este caso, la imagen sigue siendo aceptable, aunque aun sigue siendo necesario desplazar el dedo a una velocidad forzada.

La Figura 6.1, imagen *b*) muestra una imagen captura con 120 frames, donde se distingue que aproximadamente una tercera parte de la imagen pertenece a datos de solapamiento.

- **Captura 160 frames.**

El tamaño de la imagen no difiere mucho de los casos anteriores, pero este numero de frames de captura si posibilita el desplazamiento del dedo sobre la superficie del sensor dentro de unos márgenes de velocidad naturales.

La Figura 6.1, imagen *c*) muestra una imagen capturada con 160 frames, donde aproximadamente el 50% de los datos pertenecen a datos solapados.

En cualquiera de los casos la imagen resultante tiene unas dimensiones aproximadas de 232x640 pixeles.

6.2 Según la profundidad de color.

A continuación, se muestran en la Figura 6.2 imágenes con distintos niveles en la variación de la profundidad de color una vez se representa cada pixel con 8 bits en lugar de

4 bits, siendo necesario para poder ser representado y proporcionar imágenes bien contrastadas. Se parte de 16 tonos de grises, 4 bits, como el nuevo rango de grises en 8 bits es de 0 a 256 tonos, y los pixeles de los que se dispone están codificados con los 16 primeros tonos de este nuevo rango.

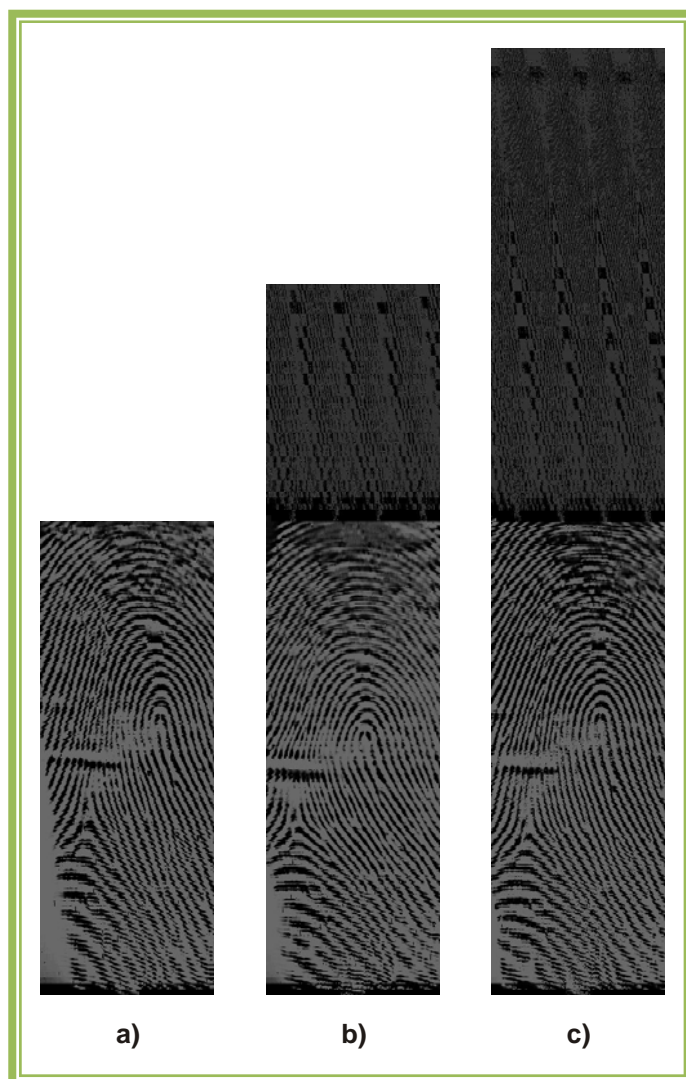


Figura 6.1: Imágenes a 80, 120 y 160 frames.

Para aprovechar todo el rango se estima oportuno multiplicar cada pixel por 0x11, así el pixel codificado con 0x0, mínimo valor de entre los posibles y equivalente al negro absoluto, queda con 0x00 y no varia; y el 0xF que es el máximo valor de los que disponemos quedaría con 0xFF que equivale a un blanco puro, los 14 valores intermedios quedarían equiespaciados en todo el rango de grises.

Para poder resolver cual es la mejor opción en cuanto al cambio de contraste de la imagen capturada, se prueban con valores intermedios, en torno a 0x07, y variando poco el nivel de contraste de la imagen original, por ejemplo 0x03.

De donde se obtiene la imagen representada en la Figura 6.2, donde la Figura 6.2, imagen *a)* corresponde al producto por 0x03, la Figura 6.2, imagen *b)* al resultado de la operación con el operando igual a 0x07, y la Figura 6.2, imagen *c)* el resultado de aprovechar todo el espectro de grises, es decir multiplicar por 0x11 dando lugar a las consideraciones tenidas en cuenta y citadas en la sección 5.1.

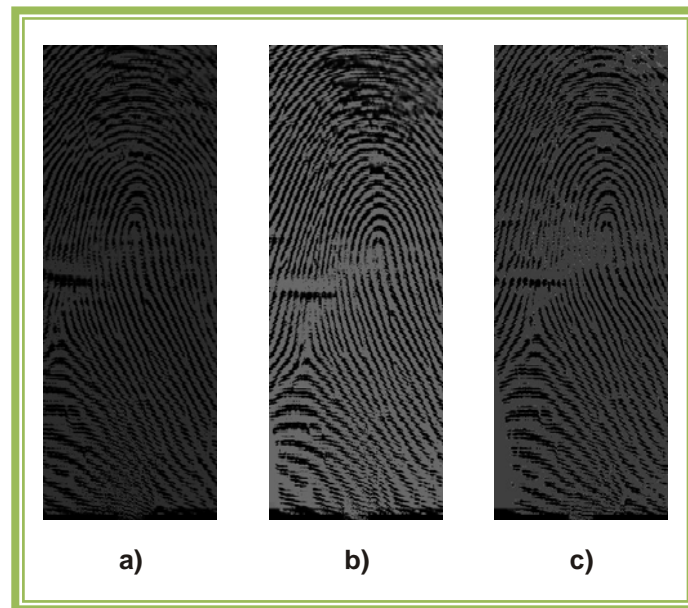


Figura 6.2: Imágenes con distintos contrastes.

6.3 Usando estabilizador de temperatura.

Se ha comprobado que en condiciones de temperatura ambiental por debajo de los 20° aproximadamente, la respuesta del sensor empeora, para intentar resolver esta situación y pretendiendo explotar las características del sensor, se prueba la adquisición de huella dactilar haciendo uso del estabilizador de temperatura con los dos valores predefinidos de 50mW y 100mW, así como desactivándolo para evaluar su contribución a la calidad de la adquisición de la huella.

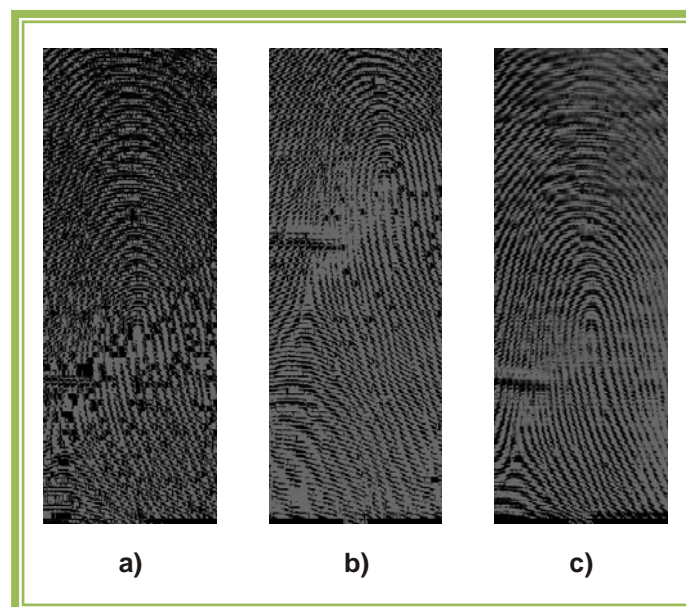


Figura 6.3: Aplicación del estabilizador de temperatura en condiciones ambientales desfavorables.

En estas condiciones se comprueba que el estabilizador de temperatura no contribuye a la mejora de la imagen, mas bien la empeora, como puede verse en la Figura 6.3, siendo la imagen *a)* la obtenida con el estabilizador de temperatura a 100 mW, la imagen *b)* el resultado de aplicar una potencia de 50 mW al estabilizador de temperatura y la imagen *c)* sin el uso del estabilizador de temperatura.

En las condiciones ambientales en las que se obtiene el mejor rendimiento del sensor, el uso de esta funcionalidad no ayuda tampoco a obtener mejores resultados como se muestra en la Figura 6.4, aplicando el mismo proceso que en la figura anterior.

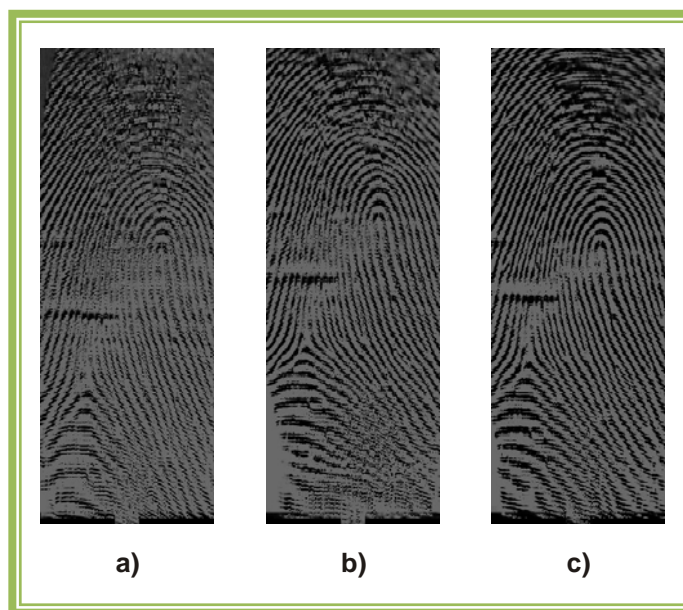


Figura 6.4: Aplicación del estabilizador de temperatura en condiciones ambientales favorables.

6.4 Otros registros del sensor.

Aparte de los utilizados para el estabilizador de temperatura otros de los registros necesarios para la configuración del sensor, son el *clickctrl*, *navctrl* y *movctrl*. Estos registros no afectan a la calidad de la imagen, sino a la sensibilidad del sensor a la detección del dedo sobre su superficie, el desplazamiento del mismo y parámetros relacionados con el desplazamiento respectivamente.

6.5 Procesando la imagen.

Una vez determinados los mejores parámetros de configuración del sensor y tratamiento de la imagen, existe la posibilidad de someter a la imagen a un pequeño proceso de mejora de calidad de imagen con objeto de reducir la distorsión producida por el desplazamiento y eliminación de ruido, para posteriormente convertir la imagen en escala de grises a blanco y negro. Para conseguirlo se aplica desde Irfanview un perfilado y un filtro de mediana, y se pasa el bpp a 1. Una secuencia de este proceso se puede ver en la Figura 6.5, la imagen *a)* muestra la huella capturada, la imagen *b)* muestra la imagen capturada aplicando un perfilado de los bordes, la imagen *c)* además aplica al resultado de *b)* un filtro de mediana y por ultimo en la imagen *d)* se tiene el resultado de aplicar el perfilado, el filtro de mediana y conversión de la imagen a blanco y negro.

El perfilado y el cambio de la imagen a blanco y negro se ejecutan en Irfanview por línea de comando desde la aplicación creada para la representación en pantalla. Sin embargo, el filtro de mediana no se puede ejecutar por línea de comando siendo necesario aplicarlo en caso de necesitarlo desde la propia aplicación Irfanview.

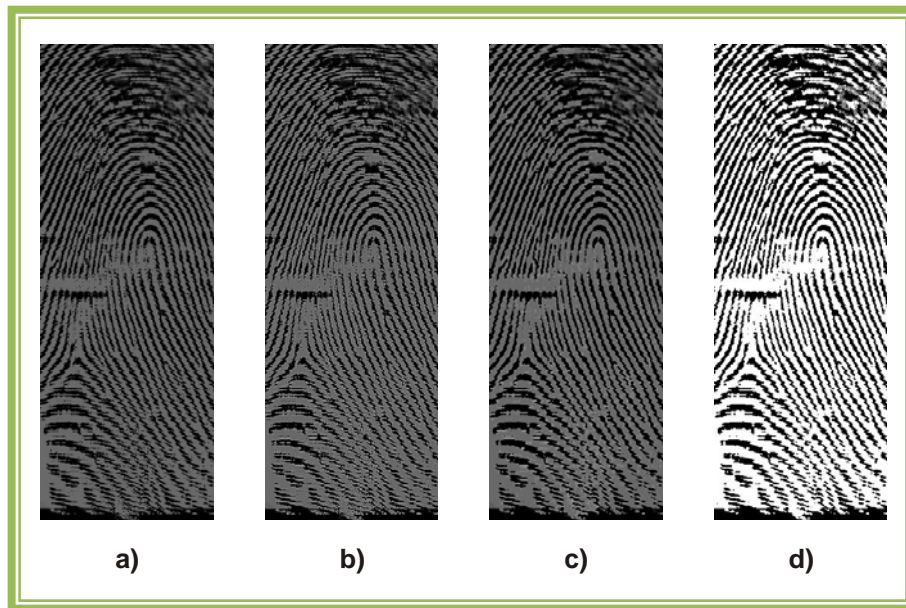


Figura 6.5: Fases del Procesado de la Imagen.

Capítulo 7

Conclusiones y Líneas Futuras

En este capítulo se presentan las principales conclusiones sobre el trabajo realizado así como las futuras líneas de trabajo.

7.1 Conclusiones

En este Proyecto Fin de Carrera, se ha utilizado el sensor de huella dactilar térmico por desplazamiento fabricado por Atmel AT77C104B para la captura de imágenes de huella dactilar como apuesta para su implantación en sistemas empotrados portátiles. Las principales conclusiones obtenidas se detallan a continuación:

La primera de ellas se refiere al enfoque térmico. Aunque la literatura consultada y la propia documentación del dispositivo hacen referencia a la resistencia de esta tecnología a condiciones ambientales extremas, en las pruebas llevadas a cabo, se ha podido comprobar que en temperatura ambiente por debajo de los 19°C, la calidad en las imágenes capturadas merma. Esto no parece achacable al dispositivo, sino a la disminución de la temperatura corporal del propio individuo al estar sometido a esta temperatura ambiental, por lo que entendiéndose que por lo general el dispositivo y el usuario están expuestos a los mismos factores ambientales (pues hay que recordar que lo que se pretende de estos sensores es la implantación en dispositivos móviles) a bajas temperaturas corporales la captura resulta de baja calidad o nula, sin que el estabilizador de temperatura ayude a corregirlo, siendo necesario aumentar la temperatura corporal para una buena respuesta del sensor, por lo que parece conveniente pensar en sensores basados en otras tecnologías para combatirlo.

Una característica, y quizá la mas ventajosa del sensor, es la integración de la circuitería para la reconstrucción de la huella en el propio circuito integrado. Esto no sólo permite un tiempo de ejecución de la operación mas rápido, pues reduce el tiempo de procesado, sino que también ahorra mucho tiempo en la parte de desarrollo, pues según lo visto en las pruebas realizadas, esta reconstrucción parece suficientemente buena, aunque esta limitada por la velocidad de desplazamiento del dedo con un mínimo de 2 cm/Sg y un máximo de 20 cm/Sg y no evita tener que tratar la imagen posteriormente para la eliminación de ruidos y distorsiones.

Finalmente y aunque no se haya probado en este estudio, porque no era parte de los objetivos del mismo, la posibilidad de utilizar el sensor como sistema de navegación en el dispositivo a modo de ratón parece un buen enfoque, por la comodidad de su posible uso y la reutilización del sensor para otras tareas.

Las líneas de trabajo futuras se presentan en el siguiente apartado.

7.2 Líneas Futuras

En este apartado, se presentan posibles mejoras y futuras líneas de trabajo que podrían complementar este proyecto.

Comprobar la utilidad del sensor en un sistema biométrico: Hasta ahora se ha considerado buena la captura y reconstrucción del sensor, esta opinión se basa en la

apreciación humana de las imágenes capturas, sabiendo que el cerebro humano esta muy bien entrenado para el reconocimiento de formas o de patrones. Parece conveniente probar las imágenes capturadas por el AT77C104 en un sistema biométrico real y evaluar su calidad.

Procesar las imágenes adquiridas con mayor rigor: En este trabajo se ha recurrido a un procedimiento de mejora de la imagen muy básico, con el objeto de poder mostrar una representación de la captura más amigable. Sin embargo parece conveniente recurrir a un procesado más riguroso de la imagen adquirida de cara a su implantación en un sistema biométrico real.

Desarrollar un procedimiento de adquisición mas flexible: Debido a las limitaciones presentadas en la sección 5.1, el método de captura de imagen de este estudio resulta muy rígido en cuanto a velocidad de desplazamiento del dedo, pues aunque se haya ajustado a velocidades de movimiento naturales, parece conveniente detectar por medio de interrupciones y gestión del espacio de memoria, el inicio del desplazamiento del dedo y la detección del fin de ese desplazamiento otorgando mayor libertad al usuario.

Desarrollar aplicación para adaptar e incorporar el protocolo PS/2: Como se ha comentado varias veces en este texto, algunos de los registros del sensor son semejantes a los utilizados en el protocolo PS/2 para un interfaz de ratón. Parece interesante hacer la migración de este protocolo a las características del sensor y evaluar la comodidad y facilidad de uso del AT77C104B como interfaz entre usuario y maquina.

Bibliografía

- [1]. J. L. Aching Samatelo, D. A. Rojas Vigo, “Reconocimiento biométrico de huellas dactilares y su implementación en DSP” Proyecto Fin de Carrera en Ingeniería Eléctrica. Universidad Nacional Mayor de San Marcos, Perú, 2005.
- [2]. J. Addepalli, A. Vasudev, “Interfacing Atmel® Fingerprint Sensor AT77C104B with Blackfin® Processors” Engineer to engineer note, 2007,
http://www.analog.com/static/imported-files/application_notes/EE.325.Rev.1.08.2007.pdf.
- [3]. Biometría,
<http://es.wikipedia.org/wiki/Biometria>.
- [4]. Sensor de huella dactilar,
http://es.wikipedia.org/wiki/Sensor_de_huella_digital.
- [5]. D. Maltoni, D. Maio, A. K. Jain, S. Prabhakar “Handbook of Fingerprint Recognition” Springer, 2003.
- [6]. Tutorial Biometría,
http://www.robotiker.com/castellano/noticias/eventos_pdf/77/HOIP_2008_TUTORIAL_BIOMETRIA.pdf.
- [7]. J. Areitio Bertolín, M. T. Areitio Bertolín, “Análisis en torno a la tecnología biométrica para los sistemas electrónicos de identificación y autenticación” Revista española de electrónica, núm. 630, pp. 56-67, 2007.
- [8]. Introducción a la biometría,
<http://www.monografias.com/trabajos43/biometria/biometria.shtml>.
- [9]. A. K. Jain, S. Z. Li, “Encyclopedia of biometrics” Springer, 2009.
- [10]. P. Bishop, “Atmel's FingerChip™ Technology for Biometric Security “ Atmel white paper,
http://www.atmel.com/dyn/resources/prod_documents/FingerChip_WhitePaper_11_12_02.pdf.
- [11]. “AT77C104B FingerChip” Atmel, 2008,
http://www.atmel.com/dyn/resources/prod_documents/doc5347.pdf.
- [12]. Keil, MCB2100 user's guide,
<http://www.keil.com/support/man/docs/mcb2100/default.htm>.
- [13]. Fujitsu, MBF310 Solid state fingerprint sweep sensor™, rev 5, 2005.

- [14]. Embest LPCEB2000 Evaluation Board,
<http://www.embedinfo.com/English/Product/LPCEB2000.asp>.
- [15]. “LPC2119/2129/2194/2292/2294 USER MANUAL” NXP Semiconductors, 2008,
http://www.nxp.com/documents/user_manual/UM10114.pdf.

Anexo A

Código de la Aplicación

```
#include <LPC22xx.H>
#include <stdio.h>
#include <rt_misc.h>

#pragma import(__use_no_semihosting_swi)

/* Definiciones de los Registros del Sensor */
#define ModeCtrlWrite 0x84
#define EnCtrlWrite 0x88
#define HeatCtrlWrite 0x8C
#define NavCtrlWrite 0x90
#define ClickCtrlWrite 0x94
#define MovCtrlWrite 0x98
#define StatusRead 0xC0
//Parámetros Registro Status
#define READERR 0x08
//Parámetros Registro MODECTRL
#define MODE_STANDBY 0x00
#define MODE_NAV 0x20
#define MODE_ACQ 0x40
//Parámetros Registro ENCTRL
#define MOVEN 0x20
#define READERREN 0x04
//Parámetros Registro NAVCTRL
#define NAVFREQ_5800 0x00
#define NAVV_01 0x08
#define CLICKV_01 0x02
//Parámetros Registro CLICKCTRL
#define CLICKFREQ_60 0x40
#define CLICKDET_11 0x18
#define CLICKCPT_20 0x05

/* Definiciones del micro */
#define MAM_MODE 2 // Para configurar memoria externa
#define MAM_FETCH 3 // Para configurar memoria externa
#define MEMMAP_BOOT_LOADER_MODE 0 // Remapeo vectores interrup. hacia bloque de inicio
#define MEMMAP_USER_RAM_MODE (1<<1) // Remapeo vectores interrup. hacia SRAM
#define ESPACIO_SRAM 0x40000 // Área total de SRAM en bytes
#define INICIO_SRAM 0x81000000 // Inicio Mapa Memoria SRAM para la Rx de datos
#define ORDENAR_SRAM 0x81012360 // Inicio Mapa Memoria SRAM para los datos ordenados
#define SALIDAS 0x00003C00 // Mascara pines Salida SSS (P0.10),FSS(P0.11), RST (P0.12) y SPICLK (P0.13)
#define SPIS 0x00000C00 // Mascara pines SSS (P0.10), FSS P0.11)
#define SPICLK 0x00002000 // Mascara pin SPICLK (P0.13)
#define SSS 0x00000400 // Mascara P0.10 SSS
#define FSS 0x00000800 // Mascara P0.11 FSS
#define RST 0x00001000 // Mascara P0.12 RST
#define DUMMY 0x00 // Dummy Byte lectura SPI
#define CR 0x0D // Carriage Return

/* Declaraciones variables */
unsigned short* pSRAM = 0; // Puntero a la SRAM para la adquisición de datos
unsigned short* pOrdenados = 0; // Puntero a la SRAM para la ordenación de datos
unsigned char lecturaSensor[2]; // Contiene los datos recibidos de los registros del sensor
unsigned char irqSensor = 1; // 0:solicitud del sensor; 1:no hay solicitud del sensor
unsigned char errorLectura = 0;
unsigned int ledOr;
unsigned int led;
int uSg1 = 5;
int uSg10 = 94;
int mSg1 = 10000;
```

```

int mSg50                                = 500000;
int mSg500                              = 5000000;

/* Declaraciones funciones */
void iniciaIRQs(void);
void iniciaPLL(void);
void iniciaSPIyGPIO(void);
void iniciaUART(void);
void iniciaSRAM(void);
void iniciaSensor(void);
void espera(int uSg);
void habilitaSSS(void);
void configuraFSS(void);
void deshabilitaSSS(void);
void adquisicionDatos(void);
void escribeRegistro(unsigned char codigoOperacion, unsigned char direccion);
void escribeSRAM(unsigned short* pDireccion, unsigned short datosSRAM);
unsigned char leeRegistro(unsigned char codigoOperacion);
unsigned short leeSRAM(unsigned short* pDireccion);
void TxUart(char ch);
void ordena(void);
void Envio(void);
void finTx(void);
void EXTINT1VectoredIRQ(void) __irq;

int main(void)
{
    unsigned long volatile debug;
    for (debug = 0; debug < 10000000; debug++) // Espera para la conexión del debugger
                                                (alrededor de 0.3 seg)
    {
    } //endf
    iniciaIRQs();
    iniciaPLL();
    iniciaSPIyGPIO();
    iniciaUART();
    iniciaSRAM();
    iniciaSensor();
    espera(uSg1);
    escribeRegistro(ModeCtrlWrite, MODE_STANDBY); // Activa oscilador
    escribeRegistro(ClickCtrlWrite, CLICKFREQ_60 | CLICKDET_11 | CLICKCPT_20); // Configura
                                                                                parámetros del
                                                                                Click
    escribeRegistro(NavCtrlWrite, NAVFREQ_5800 | NAVV_01 | CLICKV_01); // Configura parámetros
                                                                                de Navigation
    escribeRegistro(MovCtrlWrite, 0x09); // Configura parámetros
                                                                                de movimiento

    while(1)
    {
        IO1SET = 0x00FF0000;
        escribeRegistro(ModeCtrlWrite, MODE_NAV); // Modo Navigation
        escribeRegistro(EnCtrlWrite, MOVEN); // Habilita irq movimiento
        printf("\nEsperando Dedo y Desplazamiento Vertical...");
        irqSensor = 1;
        while(irqSensor) // Espera movimiento del dedo
        {
        } //endw
        IO1CLR = 0x00FF0000;
        escribeRegistro(ModeCtrlWrite, MODE_STANDBY); // Modo Standby
        escribeRegistro(ModeCtrlWrite, MODE_ACQ); // Modo Adquisición de datos
        escribeRegistro(EnCtrlWrite, READERREN); // Habilita irq error lectura
        configuraFSS();
        printf("\nAdquiriendo...");
        adquisicionDatos();
        escribeRegistro(HeatCtrlWrite, 0x00);
        escribeRegistro(ModeCtrlWrite, MODE_STANDBY);
        if(errorLectura) // Si error de lectura
        {
            printf("\nError de Lectura - Volver a Escanear");
            espera(mSg500);
        } //endif
        else
        {
            ordena();
            printf("\nEnviando Imagen...");
            espera(mSg50);
            Envio();
        }
    }
}

```

```

    }//ende
  }//endw
} //Main

/* Definición de funciones */
void iniciaIRQs() // Inicializa interrupciones
{
  VICIntSelect = 0x00000000;
  EXTMODE = 0x00000002; // Selecciona sensibilidad por flanco
  EXTPOLAR = 0x00000000; // Sensibilidad flanco de subida EINT1
  EXTINT = 0x00000002; // Borra el flag de IRQ EINT1
  VICVectAddr0 = (unsigned)EXTINT1VectoredIRQ; // Pasa la direccion de IRQ al VIC
  VICVectCntl0 = 0x0000002F; // Selecciona prioridad para la EINT1 IRQ
  VICIntEnable = 0x00008000; // Habilita EINT1 en el VIC
  EXTINT = 0x00000002; // Borra el flag de IRQ
} //iniciaIRQs

void iniciaPLL() // Inicializa PLL
{
  PLLCFG = 0x00000024;
  PLLCON = 0x00000001;
  PLLFEED = 0x000000AA;
  PLLFEED = 0x00000055;
  while(!(PLLSTAT & 0x00000400));
  PLLCON = 0x00000003;
  PLLFEED = 0x000000AA;
  PLLFEED = 0x00000055;
  VPBDIV = 0x00000011;
} //iniciaPLL

void iniciaSPIyGPIO() // Inicializa SPI y GPIO
{
  PINSEL0 = 0x20005505; // SPI0, Habilita EINT1 y UART0
  IO0DIR = SALIDAS; //Pines P0.10, P0.11, P0.12, P0.13 como salidas para
  // SSS, FSS, RST y SPICLK

  IO1DIR = 0x00FF0000; //LEDs como salidas
  IO1CLR = 0x00FF0000; //Apaga los LEDs
  IO0SET = SPIs; //Deshabilita SSS, FSS
  IO0SET = SPICLK; //Habilita reloj SPI
  IO0CLR = RST; //Deshabilita RST
  SOSPCCR = 0xFA; //200KHz @ 50 MHz velocidad SPI
  SOSPCR = 0x30; //CPOL = 0, CPHA = 1, SPI master, MSB primero, sin
  // interrupciones
} //iniciaSPIyGPIO

void iniciaUART() // Inicializa interfaz serie
{
  U0LCR = 0x83; // 8 bits, sin paridad, 1 bit stop
  U0DLL = 0x1B; // Baud rate 115200 @ 50 MHz de reloj VPB
  U0DLM = 0x00; // Baud rate 115200 @ 50 MHz de reloj VPB
  U0LCR = 0x03;
  U0FCR = 0x4F;
} //iniciaUART

void iniciaSRAM()
{
  /* setup and enable the MAM (Memory Accelerator Module)
   * 1. start change by turning of the MAM (redundant) */
  MAMCR = 0;
  /* 2. set MAM-Fetch cycle to 3 cclk as recommended for > 40MHz */
  MAMTIM = MAM_FETCH;
  /* c. enable MAM */
  MAMCR = MAM_MODE;
  /* Set databus functionality */
  PINSEL2 = 0x0F000924; //0x0DE009D0
  /* SRAM 16bit * 0x81000000 - 0x81ffffff */
  # define WST2 0 /* 0...31 */
  # define WST1 0 /* 0...31 */
  # define IDCY 0 /* 0...15 */
  /* BCFG1 = 0x1000450F; */
  BCFG1 = 0x10000400 | (WST2<<11) | (WST1<<5) | (IDCY); //0x2000FBF
  # if defined(RAM_RUN)
  /* User RAM Mode. Interrupt vectors are re-mapped to Static RAM */
  MEMMAP = MEMMAP_USER_RAM_MODE;
  # endif
  pSRAM = (unsigned short*)INICIO_SRAM; // Inicia puntero a datos Rx
} //iniciaSRAM

```

```

void iniciaSensor()                                     // Inicia el sensor
{
    IO0SET    = RST;
    espera(uSg10);                                     // Espera 10 uSg
    IO0CLR    = RST;
    espera(uSg1);                                     // Respeta TrstsU
} //end iniciaSensor

void espera(int uSg)                                    // Retardo del tiempo parámetro
{
    int i;
    for(i=0; i<uSg; i++)
    {
    } //endf
} //end espera

void habilitaSSS()                                     // Habilita Slow SPI
{
    IO0CLR    = SSS;                                  // Habilita ChipSelect
    espera(uSg1);                                     // Respeta Tssu
} //habilitaSSS

void configuraFSS()                                    // Configura y habilita SPI
{
    IO0SET    = SPIs;                                 // Deshabilita SSS, FSS
    IO0SET    = SPICLK;                               // Habilita reloj SPI
    IO0CLR    = RST;                                  // Deshabilita RST
    S0SPCCR   = 0x08;                                 // Max Velocidad SPI
    S0SPCR    = 0x30;                                 // CPOL = 0, CPHA = 1, SPI master, MSB primero,
                                                    sin interrupciones
} //end configuraFSS

void deshabilitaSSS()
{
    IO0SET    = SSS;                                  // Deshabilita ChipSelect Slow SPI
    espera(uSg1);                                     // Respeta Tssu
} //deshabilitaSSS

void escribeRegistro(unsigned char codigoOperacion, unsigned char direccion) // Escribe reg.
{
    habilitaSSS();
    S0SPDR    = codigoOperacion;                      // Envía OpCode de escritura y registro
    finTx();                                         // Espera envío OpCode
    lecturaSensor[0] = S0SPDR;                      // Borra FFlag
    S0SPDR    = direccion;                          // Envía datos
    finTx();                                         // Espera envío datos
    lecturaSensor[1] = S0SPDR;
    deshabilitaSSS();
} //escribeRegistro

unsigned char leeRegistro(unsigned char codigoOperacion) // Lee registro
{
    habilitaSSS();
    S0SPDR    = codigoOperacion;                      // Envía OpCode de lectura y registro
    finTx();                                         // Espera envío OpCode
    lecturaSensor[0] = S0SPDR;
    S0SPDR    = DUMMY;                              // lee registro
    finTx();                                         // Espera envío Byte
    lecturaSensor[1] = S0SPDR;
    return lecturaSensor[1];                        // Devuelve lectura
} //leeRegistro

void adquisicionDatos()                               // Recibe imagen del sensor
{
    unsigned long i = 0;                             // Need for cycles
    unsigned char Data = 0x00;
    unsigned short temp = 0x0000;
    ledOr      = 0x00010000;
    led        = ledOr;
    pSRAM = (unsigned short *)INICIO_SRAM;           // Inicializa puntero datos
    IO0CLR     = FSS;
    do{
        S0SPDR = DUMMY;
        finTx();
        Data = S0SPDR;
    } while((Data != 0xF0)); //endDw

    for(i=0; i < 149121; i++)                        // 74560 pixels -> 80 x 233(232+Dummy) x 8

```

```

{
    if(i%2)
    {
        temp = temp | Data;
        escribeSRAM(pSRAM, temp);
        pSRAM++; // Incrementa 1 Direccion de memoria
    } //endif
    else
    {
        temp = Data;
        temp = temp << 8;
    } //ende
    SOSPDR = DUMMY;
    finTx();
    Data = SOSPDR;
    if(!(i % 18640) | (i == 149120))
    {
        IO1SET = led;
        led = led << 1;
        led = led | ledOr;
    } //endif
    } //endif
    IO0SET = FSS; // Deshabilita Slow SPI
    espera(uSgl);
} //end adquisicionDatos

void escribeSRAM(unsigned short* pDireccion, unsigned short datosSRAM) // Escribe SRAM
{
    *pDireccion = datosSRAM;
} //escribeSRAM

unsigned short leeSRAM(unsigned short* pDireccion) //Lee SRAM
{
    return *pDireccion;
} //leeSRAM

void TxUart(char ch) // Escribe caracter Puerto serie
{
    if (ch == '\n')
    {
        while (!(U0LSR & 0x20))
        {
        } //endw
        U0THR = CR;
    } //endif
    while (!(U0LSR & 0x20));
    U0THR = ch;
} //TxUart

void ordena()
{
    unsigned short temp, temp1, temp2, temp3; // Para agrupar 2 pixel convertidos a 8
                                              // bits en una palabra de 16 bits para
                                              // almacenar en memoria

    unsigned char i, j, k;
    pOrdenados = (unsigned short *)ORDENAR_SRAM;
    pSRAM = (unsigned short *)INICIO_SRAM;
    for(i = 0; i < 160; i++) // Ordena según orientación sensor
    {
        for(j = 2; j < 4; j++)
        {
            pSRAM = (unsigned short *)INICIO_SRAM;
            pSRAM = pSRAM + j + (i*466); // Puntero a inicio datos adquiridos + j
                                          // (salvo la columna de inicio de frame) +
                                          // frame (i*466), cada frame ocupa 466
                                          // posiciones de memoria
            for(k = 0; k < 116; k++) // Primera fila (i*466) si j = 2 o quinta
                                     // fila (i*466) si j = 3 del frame
            {
                temp1 = 0;
                temp2 = 0;
                temp3 = 0;
                temp = leeSRAM(pSRAM); // Contiene P2, P1, P4, P3 si j = 2 o P6,
                                       // P5, P8, P7 si j = 3
                pSRAM = pSRAM + 2; // Incremento a siguiente columna
                temp1 = temp & 0x0F00; // Cojo pixel P(1+16j) si j = 2 o P(5+16j)
                                       // si j = 3 de cada frame
            }
        }
    }
}

```

```

temp    = leeSRAM(pSRAM);

pSRAM   = pSRAM + 2;
temp2   = temp & 0x0F00;

temp2   = temp2 >> 8;

temp3   = temp1 | temp2;
temp3   = temp3 * 0x7;
escribeSRAM(pOrdenados++, temp3);
} //endf
pSRAM   = (unsigned short *)INICIO_SRAM;
pSRAM   = pSRAM + j + (i*466);

for(k = 0; k < 116; k++)
{
    temp1 = 0;
    temp2 = 0;
    temp3 = 0;
    temp  = leeSRAM(pSRAM);

    pSRAM = pSRAM + 2;
    temp1 = temp & 0xF000;

    temp1 = temp1 >> 4;

    temp  = leeSRAM(pSRAM);

    pSRAM = pSRAM + 2;
    temp2 = temp & 0xF000;

    temp2 = temp2 >> 12;

    temp3 = temp1 | temp2;
    temp3 = temp3 * 0x7;
    escribeSRAM(pOrdenados++, temp3);
} //endf
pSRAM = (unsigned short *)INICIO_SRAM;
pSRAM = pSRAM + j + (i*466);

for(k = 0; k < 116; k++)
{
    temp1 = 0;
    temp2 = 0;
    temp3 = 0;
    temp  = leeSRAM(pSRAM);

    pSRAM = pSRAM + 2;
    temp1 = temp & 0x000F;

    temp1 = temp1 << 8;

    temp  = leeSRAM(pSRAM);

    pSRAM = pSRAM + 2;
    temp2 = temp & 0x000F;

    temp3 = temp1 | temp2;
    temp3 = temp3 * 0x7;
    escribeSRAM(pOrdenados++, temp3);
} //endf
pSRAM = (unsigned short *)INICIO_SRAM;
pSRAM = pSRAM + j + (i*466);

for(k = 0; k < 116; k++)
{
    temp1 = 0;
    temp2 = 0;

```

// Contiene P10, P9, P12, P11 si j = 2 o P14, P13, P16, P15 si j = 3
// Incremento a siguiente columna
// Cojo Pixel P(9+16j) si j = 2 o P(13+16j) si j = 3 de cada frame
// P(9+16j) si j = 2 o P(13+16j) si j = 3 pasa a ocupar la parte baja de la palabra

// Cambio a 256 grises
// Almaceno ordenado y escalado

*// Puntero a inicio datos adquiridos + j (salvo la columna de inicio de frame) + frame (i*466), cada frame ocupa 466 posiciones de memoria*
*// Segunda fila (i*466) si j = 2 o sexta fila (i*466) si j = 3 del frame*

// Contiene P2, P1, P4, P3 si j= 2 o P6, P5, P8, P7 si j = 3
// Incremento a siguiente columna
// Cojo Pixel P(2+16j) si j = 2 o P(6+16j) si j = 3 de cada frame
// P(2+16j) si j = 2 o P(6+16j) si j = 3 pasa a ocupar la parte alta de la palabra
// Contiene P10, P9, P12, P11 si j = 2 o P14, P13, P16, P15 si j = 3
// Incremento a siguiente columna
// Cojo Pixel P(10+16j) si j = 2 o P(14+16j) si j = 3 de cada frame
// P(10+16j) si j = 2 o P(14+16j) si j = 3 pasa a ocupar la parte baja de la palabra

// Cambio a 256 grises
// Almaceno ordenado y escalado

*// Puntero a inicio datos adquiridos + j (salvo la columna de inicio de frame) + frame (i*466), cada frame ocupa 466 posiciones de memoria*
*// Tercera fila (i*466) si j = 2 o séptima fila (i*466) si j = 3 del frame*

// Contiene P2, P1, P4, P3 si j = 2 o P6, P5, P8, P7 si j = 3
// Incremento a siguiente columna
// Cojo Pixel P(3+16j) si j = 2 o P(7+16j) si j = 3 de cada frame
// P(3+16j) si j = 2 o P(7+16j) si j = 3 pasa a ocupar la parte alta de la palabra
// Contiene P10, P9, P12, P11 si j = 2 o P14, P13, P16, P15 si j = 3
// Incremento a siguiente columna
// Cojo Pixel P(11+16j) si j = 2 o P(15+16j) si j = 3 de cada frame

// Cambio a 256 grises
// Almaceno ordenado y escalado

*// Puntero a inicio datos adquiridos + j (salvo la columna de inicio de frame) + frame (i*466), cada frame ocupa 466 posiciones de memoria*
*// Cuarta fila (i*466) si j = 2 o octava fila (i*466) si j = 3 del frame*


```

    temp3 = 0;
    temp = leeSRAM(pSRAM);

    pSRAM = pSRAM + 2;
    temp1 = temp & 0x00F0;

    temp1 = temp1 << 4;

    temp = leeSRAM(pSRAM);

    pSRAM = pSRAM + 2;
    temp2 = temp & 0x00F0;

    temp2 = temp2 >> 4;

    temp3 = temp1 | temp2;
    temp3 = temp3 * 0x7;
    escribeSRAM(pOrdenados++, temp3);
} //endf
} //endf
} //endf
} //ordena

void Envio()
{
    unsigned short temp, temp1, j;
    unsigned char temp2, i;
    pOrdenados = (unsigned short *)ORDENAR_SRAM;
    ledOr = 0x00010000;
    led = ledOr;
    IO1CLR = 0x00FF0000;
    printf("yz\n");
    espera(mSgl);
    for(i = 0; i < 80; i++)
    {
        for(j = 0; j < 928; j++)

        {
            temp = leeSRAM(pOrdenados++);
            temp1 = temp >> 8;
            temp2 = temp1;
            TxUart(temp2);
            temp2 = temp;
            TxUart(temp2);
        } //endf
        if(!((i+1) % 10))
        {
            IO1SET = led;
            led = led << 1;
            led = led | ledOr;
        } //endif
    } //endf
    printf("\n");
    espera(mSg50);
    printf("rs\n");
    espera(mSg50);
    printf("\nProcesando Imagen...");
    espera(mSg500);
} //Envio

void finTx()
{
    while(!(S0SPSR & 0x80))
    {
    } //endw
} //finTx

void EXTINT1VectoredIRQ() __irq
{
    if(leeRegistro(StatusRead) & READERR)
    {
        errorLectura = 1;
    } //endif
    else
    {
        irqSensor = 0;
    }
}

```

// Contiene P2, P1, P4, P3 si j = 2 o P6, P5, P8, P7 si j = 3
// Incremento a siguiente columna
// Cojo Pixel P(4+16j) si j = 2 o P(8+16j) si j = 3 de cada frame
// P(4+16j) si j = 2 o P(8+16j) si j = 3 pasa a ocupar la parte alta de la palabra
//Contiene P10, P9, P12, P11 si j = 2 o P14, P13, P16, P15 si j = 3
// Incremento a siguiente columna
// Cojo Pixel P(12+16j) si j = 2 o P(16+16j) si j = 3 de cada frame
// P(12+16j) si j = 2 o P(16+16j) si j = 3 pasa a ocupar la parte baja de la palabra
// Cambio a 256 grises
// Almaceno ordenado y escalado

// Inicio datos imagen
// Los 80 frames imagen ordenada
// 928 posiciones de memoria, cada una con 2 pixels de 8 bits, son los datos relativos a 1 frame
// Primer byte de la posición de memoria
C byte de la posición de memoria
// Realimentación de transferencia
// Fin datos imagen
// Controla fin Tx SPI
//Espera hasta fin transferencia SPI
// Control irq desde el sensor
// Detecta origen irq
// Irq por error de lectura
// Irq por movimiento

```

    }//ende
    EXTINT      = 0x00000002;           // Borra el flag de IRQ
    VICVectAddr = 0x00000000;           // Dummy byte fin de IRQ
} //EXTINT1VectoredIRQ

/* implementation of putchar (also used by printf function to output data) */
int sendchar (int ch)                  /* Write character to Serial Port */
{
    if (ch == '\n')
    {
        while (!(U0LSR & 0x20));
        U0THR = CR;                      /* output CR */
    }
    while (!(U0LSR & 0x20));
    return (U0THR = ch);
}

int getkey (void)                      /* Read character from Serial Port */
{
    while (!(U0LSR & 0x01));
    return (U0RBR);
}

extern int sendchar(int ch);

struct __FILE { int handle; /* Add whatever you need here */ };
FILE __stdout;

int fputc(int ch, FILE *f)
{
    return (sendchar(ch));
}

int ferror(FILE *f)
{
    /* Your implementation of ferror */
    return EOF;
}

void _ttywrch(int ch)
{
    TxUart(ch);
}

void _sys_exit(int return_code)
{
    label: goto label; /* endless loop */
}

```